NAIST-IS-DD1661017

Doctoral Dissertation

Improving Low-Resource Machine Translation through Syntactic and Contextual Information

Akiva Miura

March 8, 2018

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Akiva Miura

Thesis Committee:

Professor Satoshi Nakamura	(Supervisor)
Professor Yuji Matsumoto	(Co-supervisor)
Associate Professor Katsuhito Sudoh	(Co-supervisor)
Assistant Professor Graham Neubig	(Carnegie Mellon University)
Doctor Masao Utiyama	(NICT)

Improving Low-Resource Machine Translation through Syntactic and Contextual Information*

Akiva Miura

Abstract

Translation is an essential tool to communicate with foreign language speakers. However, it requires specialized knowledge, and thus expectations are heightening toward machine translation (MT), which has potential to perform manual translation tasks in an automated fashion. Nowadays many practical applications of MT translate from English into other widely spoken languages and vice versa. On the other hand, MT quality has not yet reached a practical level in many language pairs that do not include English.

The current mainstream MT frameworks are statistical MT and neural MT, which are characterized by the ability to learn to translate automatically through machine learning techniques. It has been observed that translation with models trained on larger parallel corpora can achieve higher accuracy, and usually millions of sentence pairs are required in order to produce a high quality translation. Unfortunately, readily available parallel corpora are limited for most language pairs, particularly those that do not include English, having few sentence pairs, or none at all. Moreover, the cost of manually producing a high quality parallel corpus is estimated to be in the millions of dollars.

In this thesis, we focus on improving MT quality with two types of approaches for coping with the scarceness of bilingual corpora (1) *pivot translation* and (2) *active learning* for MT.

Pivot translation is a useful method for translating between languages with little or no parallel data by utilizing parallel data in an intermediate (pivot) language such as English. Although various methods using pivot languages have

*Doctoral Dissertation, Graduate School of Information Science,
Nara Institute of Science and Technology, NAIST-IS-DD1661017, March 8, 2018.

been proposed, ambiguity due to expressions in the pivot language often causes incorrect selection of translation rules and harms translation quality. Therefore, *pivot-side disambiguation* is a key issue in pivot translation. In the first part of the thesis, we propose two new pivot translation methods to solve the two types of ambiguity respectively. The first method is proposed to solve semantic ambiguity, and lets MT models remember the information of the pivot phrase. This information can help to select appropriate translation rules considering pivot-side context with pivot language models. The second method is proposed to solve syntactic ambiguity, and introduces an explicitly syntax-aware matching condition to find correct correspondence of source-pivot and pivot-target translation rules, and can produce more reliable models. Experimental results on multilingual translation show a significant improvement in all the tested language pairs.

Active learning is a framework that makes it possible to efficiently train statistical models by selecting informative examples from a pool of unlabeled data. Previous work has found this framework effective for MT, making it possible to train better MT models with less effort, particularly when annotators translate short phrases instead of full sentences. However, previous methods for phrase-based active learning for MT fail to consider whether the selected units are coherent and easy for human translators, and also have problems with selecting redundant phrases with similar content. In this part, we propose two new methods for selecting more *syntactically coherent* and *less redundant* segments in active learning for MT. Experiments using both simulation and extensive manual translation by professional translators find the proposed method effective, achieving both greater gain of translation score for the same number of translated words, and allowing translators to be more confident in their translations.

Our experiments demonstrate that MT quality can significantly benefit from syntactic and contextual information when faced with limited training data.

Keywords:

pivot translation, active learning, machine translation, parallel corpus, low-resource language pairs

Acknowledgements

ὁ Θεὸς γάρ ἐστιν ὁ ἐνεργῶν ἐν ὑμῖν καὶ τὸ θέλειν
καὶ τὸ ἐνεργεῖν ὑπὲρ τῆς εὐδοκίας.

(ΠΡΟΣ ΦΙΛΙΠΠΗΣΙΟΥΣ 2:13)

『神は御意を成さんために汝らの衷にはたらき、
汝等をして志望をたて、業を行はしめ給へばなり。』
(ピリピ人への書 2 章 13 節　大正改訳)

私のうちに志と使命を与え，今日まで導いてくださった主イエス・キリストの御名を讃えます．本当に多くの方々に助けられ，支えられながら博士論文の執筆に至ることができました．

本学情報科学研究科の中村哲教授には，博士前期・後期課程を通じての 4 年間，指導教官として常に熱心なご教鞭をいただきました．入学時から希望していた研究テーマに挑戦させていただき，なかなか成果が出ない時にも辛抱強くご指導いただきました．また，身に余るほどのご支援をいくつも承りました．心より感謝いたします．

本学情報科学研究科の松本裕治教授には，副指導教官として大変貴重なご指導とご助言をいただきました．心より感謝いたします．

本学情報科学研究科の須藤克仁准教授 (前 NTT コミュニケーション科学基礎研究所) には，副指導教官として熱心なご教示とご助言をいただきました．NTT の研究所にご在籍されていた頃より関西機械翻訳勉強会をはじめ，様々な会合でお会いしては研究者人生として興味深いお話を聞かせていただきました．特に本学に異動して来られてからは多くのご助言をいただき，前進することができました．心より感謝いたします．

Carnegie Melon University の Graham Neubig 助教 (前 奈良先端科学技術大学院大学 助教) には，副指導教官として熱心なご教示とご助言をいただきました．入学時より研究テーマに全面的に熱意をもってご指導いただき，全く

未経験であった論文執筆や研究発表に対しても丁寧にご助言いただきました．本学で執筆したすべての論文を最初に添削してくださり，とりわけ，国際会議に複数の英語論文を投稿し, 採択されたのは先生のご指導無しには成し得ませんでした．私の在学中に，本学より Carnegie Melon University へ異動されたため，4 年間のうち約 2 年間は，遠隔地からご指導いただきましたが，毎週のように個別でミーティングを行っていただき，手厚いご指導を承りました．心より感謝いたします．

情報通信研究機構の内山将夫様には，副審査委員として大変貴重なご指導とご助言をいただきました．心より感謝いたします．

名古屋大学情報科学研究科の戸田智基教授 (前 奈良先端科学技術大学院大学 准教授) には，本学勤務時において，研究全般にわたり貴重なご助言をいただきました．心より感謝いたします。

本学情報科学研究科の鈴木優特任准教授，Sakriani Sakti 特任准助教，吉野幸一郎助教，田中宏季助教には，研究室ゼミナールや研究班ミーティング等において貴重なご助言をいただきました．心より感謝いたします．

知能コミュニケーション研究室秘書の松田真奈美様には，研究室の活動を通して様々な面でお世話になりました．心より感謝いたします．

株式会社 ATR-Trek の Michael Paul 様，中澤聡様，中村明様には，機械翻訳のための能動学習手法に関する研究において，活発な議論の場を設け数々の有益なご助言をいただいたことで，研究を深めることができました．心より感謝いたします．

首都大学東京の小町守准教授には，私が機械翻訳の研究を願って進路に悩んでいた際に，メールで相談に乗ってくださり，本学への入学を勧めてくださりました．これが大きなきっかけとなり，本学の博士前期課程へ入学し，後に博士後期課程へと進学しましたが，お聞きしていた情報の通り，本当に恵まれた環境で学業に邁進することができました．情報処理学会の自然言語処理研究会で発表を行った際には，毎回質問してくださったり，研究室にお呼びいただき学生達との議論の場を用意してくださったりもしました．心より感謝いたします．

情報通信研究機構の藤田篤様には，私の研究に関心を寄せていただき，学会や勉強会でお会いしては声をかけてくださったり，様々な面でお世話になる機会がありました．洞察が鋭く，的確なご意見をいただく度に，新しい着眼点を得ることができ，また，研究への取り組みも正されました．心より感

謝いたします．

最後になりますが，私が 18 歳で実家を離れてからも暖かく支え続けてくれた母・恵子と，修士論文執筆直前に急逝した父・純二には，返しきれない恩を感じつつ，心より感謝いたします．妻・彩華には，博士後期課程の 2 年間，苦しい日々が続く中，常に支え続けてくれました．心より感謝いたします．

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Language is a key communication tool for human beings, and also signifying group identity deeply rooted in social and cultural background. Translation is an essential tool to communicate with foreign language speakers. However, it requires specialized knowledge, and thus expectations are heightening toward machine translation (MT), which has potential to perform translation tasks in an automated fashion. Nowadays many practical applications of MT translate from English into other widely spoken languages and vice versa. On the other hand, MT quality has not yet reached a practical level in many language pairs that do not include English. Therefore, it is hard to say that users who are not familiar with English can use MT between various languages without difficulty.

1.2 Known Issues in Multilingual Machine Translation

The most traditional framework of MT is Rule-Based Machine Translation (RBMT (Nirenburg, 1989)), which is implemented by manual description of translation rules. This requires knowledge of experts familiar with both of the source and target languages, and thus it is difficult to cover a wide variety of expressions in many language pairs. Therefore, this thesis discusses the framework of Statistical Machine Translation (SMT (Brown et al., 1993)), which can automatically obtain translation rules from given bilingual corpora (also referred to as "parallel corpora"), consisting of a set of sentences in two languages, through machine learning. It has been observed that translation with models trained on larger parallel corpora can achieve higher accuracy, and usually millions of sentence pairs

are required in order to produce a high quality translation system (Dyer et al., 2008; Koehn et al., 2007). Such large bilingual corpora can be obtained for a few language pairs through the on-going translation efforts of organizations such as the Canadian parliament (English-French), the United Nations (6 official languages, including Arabic and Chinese), and the European Parliament (spanning 21 European languages).

Unfortunately, readily available parallel corpora are limited for most other language pairs, particularly those that do not include English, having fewer than 100k sentence pairs, or none at all. Moreover, the cost of manually producing a high quality parallel corpus is estimated to be in the millions of dollars. For example, Germann (2001) estimated the cost of hiring professional translators to create a Tamil-English corpus at $0.36 per word, or in other words, $1.44M for 200k sentences with a 20 words per sentence average.

Indeed, this scarcity of parallel corpora makes it difficult to construct reasonably performing MT systems for most language pairs, a problem which has therefore received attention by both researchers and industries and is referred to as "low-resource machine translation" (Irvine and Callison-Burch, 2013; Lopez and Post, 2013). Advances in this field could facilitate communication across cultures, enable faster commercial expansion to new growing markets, and may even assist during disaster relief operations (Munro, 2013).

1.3 Approaches toward Low-Resource Machine Translation

As mentioned above, it is not practical to create a bilingual corpus in a straightforward manner from the viewpoint of budget and time cost. There are two possible approaches to MT between low-resource language pairs.

Effective use of indirectly available data: One approach is an indirect method reusing existing data to realize translation in the intended language pairs even if no direct source-target parallel corpus is available. An effective and representative method of this approach is to introduce a *pivot language* for which parallel data with the source and target languages exists (*pivot translation*)

(de Gispert and Mariño, 2006).

For example, since there is almost no large Japanese-French bilingual corpus (e.g. of more than 1M sentence pairs), it is difficult to directly train MT models. However, there are many large-scale and well-maintained corpora available for Japanese-English and English-French respectively. In many cases, each of the widely spoken languages other than English, like Japanese and French in this example, has a sufficiently large bilingual corpus with English. Therefore MT between these languages via English as a pivot is possible and becomes a realistic solution.

Although various methods using pivot languages have been proposed (Cohn and Lapata, 2007; Utiyama and Isahara, 2007), ambiguity due to expressions in the pivot language often causes incorrect selection of translation rules and harms translation quality. Therefore, *pivot-side disambiguation* is a key issue in pivot translation.

Efficient construction of bilingual corpus: While in specific cases large corpora can be collected, for example by crawling the web (Resnik and Smith, 2003), in many domains or language pairs it is still necessarily to create data by hand, either by hiring professionals or crowdsourcing (Zaidan and Callison-Burch, 2011). In theses cases, *active learning*, which selects which data to annotate based on their potential benefit to the translation system, has been shown to be effective to improving SMT systems while keeping the required amount of annotation to a minimum (Ananthakrishnan et al., 2010a; Eck et al., 2005; González-Rubio et al., 2012; Green et al., 2014; Haffari and Sarkar, 2009; Haffari et al., 2009; Turchi et al., 2008).

Most work on active learning for SMT assigns priority to sentences or subsentences that contain data that is potentially useful to the MT system according to a number of criteria. Sub-sentential annotation methods can remove many *redundant segments* duplicated in sentences to be selected in full-sentential annotation and reduce annotation workload theoretically. However, selected segments are not always easy to translate for humans, causing actual results to be far from the optimal. For example, *fragments* of complex phrases that have only incomplete syntactic information may harm the annotation quality. Therefore, it will

be necessary to consider methods that are truly efficient for human annotators.

1.4 Thesis Scope

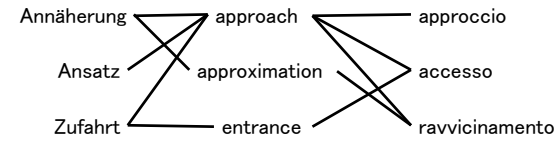
The research purpose of this thesis is to improve multilingual SMT, specifically between low-resource language pairs, resolving the known problems of previously proposed methods in pivot translation and active learning for SMT. This thesis addresses the problems to disambiguate in pivot translation through pivot-side contextual and syntactic information, and to provide more efficient and human-friendly active learning method emphasizing new criteria of non-redundancy and syntactic coherence.

1.4.1 Better Pivot Translation by Remembering the Pivot

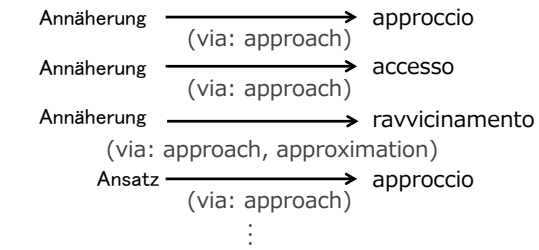
Among various methods using pivot languages, the triangulation method (Cohn and Lapata, 2007; Utiyama and Isahara, 2007), which translates by combining source-pivot and pivot-target translation models into a source-target model, has been shown to be one of the most effective approaches. However, word sense ambiguity and interlingual differences of word usage cause difficulty in accurately learning correspondences between source and target phrases.

Figure 1.1 (a) shows an example of three words in German and Italian that each correspond to the English polysemic word "approach." In such a case, finding associated source-target phrase pairs and estimating translation probabilities properly becomes a complicated problem. Furthermore, in the conventional triangulation method, information about pivot phrases that behave as bridges between source and target phrases is lost after learning phrase pairs, as shown in Figure 1.1 (b).

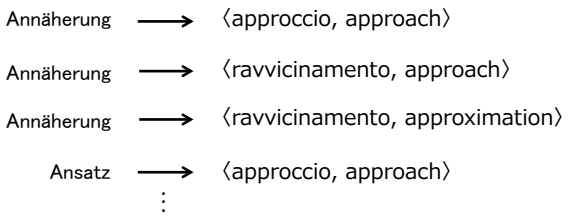
To overcome these problems, we propose a novel triangulation method that *remembers* the pivot phrase connecting source and target in the records of phrase/rule table, and estimates a joint translation probability from the source to target and pivot simultaneously. We show an example in Figure 1.1 (c). The advantage of this approach is that generally we can obtain rich monolingual resources in pivot languages such as English, and SMT can utilize this pivot-side contextual information to improve the translation quality.



(a) Triangulation (De-En-It)



(b) Traditional Triangulated Phrases

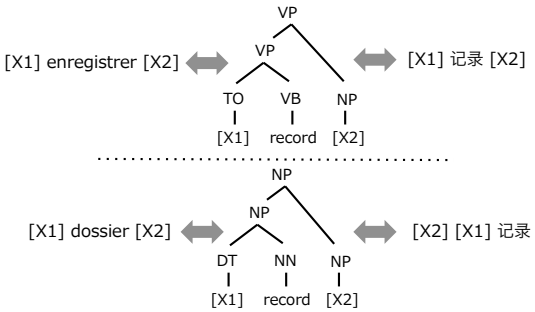


(c) Proposed Triangulated Phrases

Figure 1.1: An example of (a) triangulation and the resulting phrases in the (b) traditional method of forgetting pivots and (c) our proposed method of remembering pivots.

To utilize information about the pivot language at translation time, we train a Multi-Synchronous Context-free Grammar (MSCFG) (Neubig et al., 2015), a generalized extension of Synchronous CFGs (SCFGs) (Chiang, 2007), that can generate strings in multiple languages at the same time. To create the MSCFG, we triangulate source-pivot and pivot-target SCFG rule tables not into a single source-target SCFG, but into a source-target-pivot MSCFG rule table that remembers the pivot. During decoding, we use language models (LMs) over both the target and the pivot to assess the naturalness of the derivation.

(a) Standard triangulation method matching phrases



(b) Proposed triangulation method matching subtrees

Figure 1.2: Example of disambiguation by parse subtree matching (Fr-En-Zh), [X1] and [X2] are non-terminals for sub-phrases.

1.4.2 Syntactic Matching Methods in Pivot Translation

In the triangulation method, source-pivot and pivot-target phrase pairs are connected as a source-target phrase pair when a common pivot-side phrase exists. In Figure 1.2 (a), we show an example of standard triangulation on Hiero (Chiang, 2007) translation models that combines hierarchical rules of phrase pairs by matching pivot phrases with equivalent surface forms. This example also demonstrates problems of ambiguity: the English word "record" can correspond to several different parts-of-speech according to the context. More broadly, phrases including this word also have different possible grammatical structures, but it is impossible to uniquely identify this structure unless information about the surrounding context is given.

This varying syntactic structure will affect translation. For example, the French verb "enregistrer" corresponds to the English verb "record", but the French noun "dossier" also corresponds to "record" — as a noun. As a more extreme example, Chinese is a languages that does not have inflections according to the part-of-speech of the word. As a result, even in the contexts where "record" is used

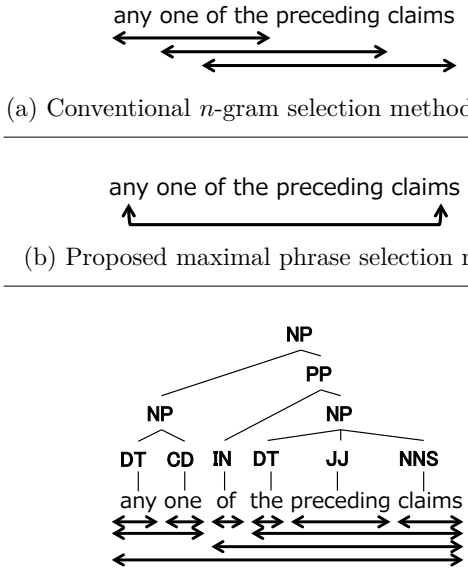
with different parts-of-speech, the Chinese word "记录" will be used, although the word order will change. These facts might result in an incorrect connection of "[X1] enregistrer [X2]" and "[X2] [X1] 记录" even though proper correspondence of "[X1] enregistrer [X2]" and "[X1] dossier [X2]" would be "[X1] 记录 [X2]" and "[X2] [X1] 记录". Hence a superficial phrase matching method based solely on the surface form of the pivot will often combine incorrect phrase pairs, causing translation errors if their translation scores are estimated to be higher than the proper correspondences.

Given this background, we hypothesize that disambiguation of these cases would be easier if the necessary syntactic information such as phrase structures are considered during pivoting. To incorporate this intuition into our models, we propose a method that considers syntactic information of the pivot phrase, as shown in Figure 1.2 (b). In this way, the model will distinguish translation rules extracted in contexts in which the English symbol string "[X1] record [X2]" behaves as a verbal phrase, from contexts in which the same string acts as noun phrase.

1.4.3 Syntactic and Non-Redundant Segment Selection for Active Learning

Most work on active learning for SMT, and natural language tasks in general, has focused on choosing which *sentences* to give to annotators. These methods generally assign priority to sentences that contain data that is potentially useful to the MT system according to a number of criteria. For example, there are methods to select sentences that contain phrases that are frequent in monolingual data but not in bilingual data (Eck et al., 2005), have low confidence according to the MT system (Haffari et al., 2009), or are predicted to be poor translations by an MT quality estimation system (Ananthakrishnan et al., 2010a). However, while the selected sentences may contain useful phrases, they will also generally contain many already covered phrases that nonetheless cost time and money to translate.

To solve the problem of wastefulness in full-sentence annotation for active learning, there have been a number of methods proposed to perform *sub-sentential annotation of short phrases* for natural language tasks (Bloodgood and Callison-



(a) Conventional n-gram selection method ($n = 4$)

(b) Proposed maximal phrase selection method

(c) Proposed parse subtree selection method

Figure 1.3: Conventional and proposed data selection methods

Burch, 2010; Settles and Craven, 2008; Sperber et al., 2014; Tomanek and Hahn, 2009). For MT in particular, Bloodgood and Callison-Burch (2010) have proposed a method that selects poorly covered n-grams to show to translators, allowing them to focus directly on poorly covered parts without including unnecessary words. Nevertheless, our experiments identified two major practical problems with this method. First, as shown in Figure 1.3 (a), many of the selected phrases overlap with each other, causing translation of *redundant* phrases, damaging efficiency. Second, it is common to see *fragments* of complex phrases such as "one of the preceding," which may be difficult for workers to translate into a contiguous phrase in the target language.

In this work, we propose two methods that aim to solve these two problems and improve the efficiency and reliability of segment-based active learning for SMT. For the problem of overlapping phrases, we note that by merging overlapping phrases, as shown in Figure 1.3 (b), we can reduce the number of redundant words annotated and improve training efficiency. We adopt the idea of *maximal substrings* (Okanohara and Tsujii, 2009; Yamamoto and Church, 2001) which

both encode this idea of redundancy, and can be calculated to arbitrary length in linear time using enhanced suffix arrays. For the problem of phrase structure fragmentation, we propose a simple heuristic to count only *well-formed syntactic constituents* in a parse tree, as shown in Figure 1.3 (c).

1.5 Document Structure

This thesis contains the following chapters:

- In Chapter 2, the four representative frameworks of MT in historical order are introduced. Then, the detailed mechanism of SMT is explained, and also a basic summary of MT framework based on Neural Networks is provided.
- In Chapter 3, conventional methods in pivot translation and active learning for SMT are described. Specifically their own problems and potential methods to solve them are discussed.
- In Chapter 4, a proposed method in pivot translation to resolve pivot-side *contextual* ambiguity is presented. This proposed method lets MT models remember the information of the pivot phrase. This information can help to select appropriate translation rules considering pivot-side context with pivot language models. This chapter is based on an ACL'2015 paper (Miura et al., 2015).
- In Chapter 5, a proposed method in pivot translation to resolve the pivot-side *syntactic* ambiguity is presented. This proposed method introduces an explicitly syntax-aware matching condition to find correct correspondence of source-pivot and pivot-target translation rules, and can produce more reliable models. This chapter is based on a WMT'2017 paper (Miura et al., 2017).
- In Chapter 6, a proposed method in active learning for SMT to introduce new criteria of segment selection, based on *non-redundancy* and *syntactic coherence*. This proposed method provides more compact and human-friendly annotation task than conventional methods, resulting in a higher

quality parallel corpus with lower annotation cost. This chapter is based on an NAACL'2016 paper (Miura et al., 2016).

- In Chapter 7, summary and contributions of this thesis are described, and directions for future work are discussed.

2 Machine Translation Frameworks

Machine Translation (MT) is a computer-aided technology converting sentences of a certain language into sentences of different languages. To realize MT, various frameworks have been proposed in history, and as representative there are Rule-Based Machine Translation (RBMT (Nirenburg, 1989)[1]), Example-Based Machine Translation (EBMT (Nagao, 1984))[2] Statistical Machine Translation (SMT (Brown et al., 1993)) and Neural Machine Translation (NMT (Bahdanau et al., 2015; Sutskever et al., 2014)).

Since translation rules are manually described in RBMT, it has an advantage that translation results can be controlled based on linguistically motivated grammar rules. On the other hand, knowledge of experts familiar with both of source and target languages is required for each language pair, it is necessary to describe a huge and complicated variety of rules, and it is difficult to cover a variety of language expressions. In the other three frameworks, there are many common points, such as automatically acquiring translation rules based on bilingual corpora.

In EBMT, a target language sentence is generated by combining examples having a high degree of similarity with respect to an input sentence based on the example database of parallel sentences. This framework is known to exhibit extremely high performance when the bilingual corpus used for examples and the field of the sentence to be translated conform to each other. However, it is necessary to consider complicated combination problems, statistical models are not assumed for scoring at the time of selecting examples, and thus lack versatility.

[1]RBMT is referred also as "Knowledge-Based Machine Translation (KBMT)."
[2]EBMT is referred also as "Machine Translation by Principal Analogy."

SMT, which is discussed centrally in this thesis, automatically acquires correspondence relationships of words and phrases from bilingual corpora as translation rules, gives probabilistic scores to each rule. Then, for each input sentence, it searches for an output sentence that maximizes the translation probability score for given input sentence. Depending on the translation rules to be used, there is a more detailed classification of the SMT frameworks, and these are introduced in Sections 2.1.3-2.1.7. Generally the versatility of frameworks based on statistical models is high, and generalized log-linear models can be used for efficient search and optimization (Och, 2003). Since it has become possible to operate large-scale computing resources and rich language resources in recent years, SMT research and development is spurred and many of MT systems that adopt SMT.

Research on NMT, which implements MT using neural networks (NNs), also tends to increase in recent years, and various methods have been devised. In some cases, NNs are used to strengthen SMT models, whereas there are many methods to train translation models directly from parallel corpora. It has many attractive advantages that are difficult with ordinary SMT, such as training of long distance dependencies, joint optimization of multiple models, etc. However, problems remain in practical application, including the fact that advanced parallel computing environments are indispensable, and interpretation and control of trained models is difficult.

Although each framework has advantages and disadvantages, this thesis focuses on SMT, and only some experiments have comparison and verification with NMT. As the reason for using SMT in this study, specifically since automatically acquired translation rules are scored based on statistical model, it is possible to explicitly give meanings when synthesizing models, and therefore, SMT is more easily integrated with pivot translation and active learning. It has also reported that NMT can not exert much performance and is inferior in accuracy of SMT when bilingual corpus used for training is not sufficient. Therefore, SMT should be more suitable for low-resource scenarios.

The following sections describe about detailed mechanism of SMT (Section 2.1) and explain briefly about the representative mechanisms of NMT (Section 2.2).

2.1 Statistical Machine Translation

The basic idea of SMT is based on a noisy channel model (Shannon, 1948). For given sentence $\boldsymbol{f}$ in source language, let $\mathcal{E}(\boldsymbol{f})$ be a set of all the possible translated sentences in target language. Assume that $Pr(\boldsymbol{e}|\boldsymbol{f})$, translation probability from $\boldsymbol{f}$ to $\boldsymbol{e}$, can be computed for any sentence $\boldsymbol{e}$ in target language. SMT searches for the $\hat{\boldsymbol{e}} \in \mathcal{E}(\boldsymbol{f})$ that maximizes $Pr(\boldsymbol{e}|\boldsymbol{f})$, having maximal translation probability in target language.[1]

$$\begin{aligned}
\hat{\boldsymbol{e}} &= \arg\max_{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})} Pr(\boldsymbol{e}|\boldsymbol{f}) && (2.1)\\
&= \arg\max_{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})} \frac{Pr(\boldsymbol{f}|\boldsymbol{e})Pr(\boldsymbol{e})}{Pr(\boldsymbol{f})} && (2.2)\\
&= \arg\max_{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})} Pr(\boldsymbol{f}|\boldsymbol{e})P(\boldsymbol{e}) && (2.3)
\end{aligned}$$

However, since it is difficult to implement this actual model accurately, it is common to reformulate as a weight optimization problem based on the following log-linear model (Och, 2003).

$$\begin{aligned}
\hat{\boldsymbol{e}} &= \arg\max_{\boldsymbol{e}\in\mathcal{E}(f)} Pr(\boldsymbol{e}|\boldsymbol{f}) && (2.4)\\
&\approx \arg\max_{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})} \frac{\exp\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{f},\boldsymbol{e}\right)}{\sum_{e'}\exp\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{f},\boldsymbol{e'})\right)} && (2.5)\\
&= \arg\max_{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})} \boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{f},\boldsymbol{e}) && (2.6)
\end{aligned}$$

where $\boldsymbol{h}$ is a feature vector, with feature values such as logarithmic probability scores of translation models (Section 2.1.1) and language models (Section 2.1.2), word reordering score accompanying derivation, and various penalties. The exact number and type of features varies from framework to framework. $\boldsymbol{w}$ is a vector of parameters, having the same dimension number with $\boldsymbol{h}$, that weights each element of the feature vector. To adjust each element of $\boldsymbol{w}$ to be optimum, it is necessary to use development data (referred also as tuning data) obtained by

[1] Such a process that searches for the optimal translation candidate maximizing the translation probability is referred as decoding.

separating a bilingual corpus from training data and testing data, and automatic evaluation measure that computers the similarity score of translation result with the reference sentence in target language. Then the optimizer finds parameters such that the evaluation score such as BLEU (Papineni et al., 2002) is maximized for the input sentences (Och, 2003). The various SMT frameworks described in Sections 2.1.3-2.1.7 are also based on this log-linear model, but use different features.

2.1.1 Translation Models

Translation model (TM) $Pr(\boldsymbol{f}|\boldsymbol{e})$ is a statistical model for prescribing the likelihood of a translation, and trained from bilingual corpus. Translation models do not directly associate $\boldsymbol{e}$ and $\boldsymbol{f}$, and assume that $\boldsymbol{f}$ is generated from $\boldsymbol{f}$ through some steps, referred as derivation $\boldsymbol{d}$. In statistical models, the derivation is treated as latent variables, and Equation 2.3 is rewritten according to the following equations:

$$\begin{aligned} \hat{\boldsymbol{e}} &= \arg\max_{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})} Pr(\boldsymbol{f}|\boldsymbol{e})Pr(\boldsymbol{e}) \\ &= \arg\max_{\boldsymbol{e}\in\mathcal{E}(\boldsymbol{f})} \sum_{\boldsymbol{d}\in\mathcal{D}(\boldsymbol{f},\boldsymbol{e})} Pr(\boldsymbol{f},\boldsymbol{d}|\boldsymbol{e})Pr(\boldsymbol{e}) \end{aligned} \tag{2.7}$$

where $\mathcal{D}(\boldsymbol{f},\boldsymbol{e})$ is the set of derivations given $\boldsymbol{f}$ and $\boldsymbol{e}$.

The IBM model is known as a specific example of derivation, that associates $\boldsymbol{f}$ and $\boldsymbol{e}$ by word correspondence between the language pair, and maximizes the translation probability (Brown et al., 1993). In IBM models, $Pr(\boldsymbol{f}|\boldsymbol{e})$ is defined as a statistical model that generates $\boldsymbol{f}$ from $\boldsymbol{e}$ on word by word through word alignment $\boldsymbol{a}$.

$$Pr(\boldsymbol{f}|\boldsymbol{e}) = \sum_{\boldsymbol{a}} Pr(\boldsymbol{f},\boldsymbol{a}|\boldsymbol{e}) \tag{2.8}$$

Training word alignment is regarded as a problem of finding the $\hat{\boldsymbol{a}}$ that maximizes the conditional probability of translation model.

$$\hat{\boldsymbol{a}} = \arg\max_{\boldsymbol{a}} Pr(\boldsymbol{f},\boldsymbol{a}|\boldsymbol{e}) \tag{2.9}$$

Word alignment is expressed as a set of position pairs (j, i) representing word correspondence $\langle f_j, e_i \rangle$ in sentence pair $\langle \boldsymbol{f}, \boldsymbol{e} \rangle$.

$$\boldsymbol{a} = \{\cdots, (j, i), \cdots\} \tag{2.10}$$

Such word-oriented translation is valid for pairs of languages that are similar enough to make sense only by replacing words. However, it is insufficient since there are many cases where words do not correspond one-to-one in reality and it is difficult to consider word order. Phrase-Based Machine Translation (PBMT (Koehn et al., 2003)) dramatically improved translation accuracy than word-based SMT, using correspondence of consecutive word strings extracted from the trained word alignment. However, translation is difficult for language pairs with significantly different word orders due to complicated reordering problems in PBMT, then SMT frameworks based on tree structure has been also proposed to deal with such advanced reordering problems. Sections 2.1.3-2.1.7 describe these frameworks in detail.

2.1.2 Language Models

Language model (LM) $Pr(\boldsymbol{e})$ is used to evaluate how natural and fluent the word sequence of a given sentence is in the target language. A good language model accurately gives high probability to natural sentences and low probability to unnatural sentences as well. By referring to this information in the translation process, SMT can select more natural sentences from translation candidates, leading to more fluent translation results. This subsection describes the n-gram language models widely used in SMT.

First, assume that the naturalness of target language $\boldsymbol{e}$ sentence can be computed by probability chain as follows:

$$P(e_1^I) = \prod_{i=1}^{I+1} P_{ML}(e_i | e_0^{i-1}) \tag{2.11}$$

where I is the length of $\boldsymbol{e}$ and assume that $\boldsymbol{e} = e_1^I = e_1 \cdots e_I$. $e_0 = \langle s \rangle$ represents the start-of-sentence symbol and $e_{I+1} = \langle /s \rangle$ represents the end-of-sentence

symbol. Each conditional probability can be obtained using maximum likelihood estimation as follows:

$$P_{ML}(e_i|e_0^{i-1}) = \frac{c_{train}(e_0^i)}{c_{train}(e_0^{i-1})} \tag{2.12}$$

where c_{train} represents the occurrence count of the word string in the training data.

However, this definition gives probability 0 to a sentence which does not appear in the training data, and can not determine the superiority or inferiority of naturalness evaluation for many translation candidates that are evaluated with score 0. Therefore, to calculate the conditional probability of e_i, it is better to use the conditional probability that considers only the e_{i-n+1}^{i-1}, immediately preceding $n-1$ words, not e_0^{i-1}, all the word string before e_i. According to this idea, Equation 2.11 is rewritten as following:

$$P(e_i^I) \approx \prod_{i=1}^{I+1} P_{ML}(e_i|e_{i-n+1}^{i-1}) \tag{2.13}$$

The n-gram language models trained in this manner, can give probability scores to sentences that do not exist in training data as well. However, even in this equation, there still remains the problem of estimating probability 0 for sentences containing n-gram that do not appears in training data. To solve this problem, there are smoothing methods that estimate the probability score by combining the conditional probabilities $P(e_i|e_{i-n+1}^{i-1})$ of n-gram and $P(e_i|e_{i-n+2}^{i-1})$ of $(n-1)$-gram. Various methods have been proposed for smoothing, and the representative methods are linear interpolation and the Kneser-Ney method (Chen and Goodman, 1996).

2.1.3 Phrase-Based SMT

PBMT is the most representative method in SMT. The training process of PBMT translation models first trains the word alignment from bilingual corpus, extracts phrase pairs based on the trained word alignment, and scores each phrase pair.

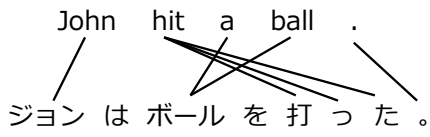


Figure 2.1: En-Ja word alignment

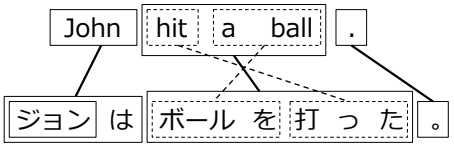


Figure 2.2: En-Ja phrase extraction

Figure 2.1 shows an example of English-Japanese word alignment obtained from bilingual corpus by the method described in Section 2.1.1.[2] Figure 2.2 shows an example of phrase extraction by finding correspondence of phrases from the obtained word-alignment. As shown in the figure, the lengths of extracted phrases are not uniquely determined, and multiple phrase pairs with different lengths are extracted according to the word alignment. However, the extracted phrase pairs impose a constraint that there is no word alignment that traverses inside and outside the two corresponding phrases, and the maximum phrase length also should be limited. PBMT translation models are trained by counting occurrence count of phrases and co-occurrence count of phrase pairs based on all the listed phrase pairs extracted in this manner.

Unlike the word-based translation models, the phrase-based translation models use the extracted phrases as the basic units of translation, thereby efficiently training translation rules of consecutive word strings such as idioms, achieving higher quality of translation. Depending on how phrases are delimited, there are multiple candidates for derivations that derivations that result in a given sentence to be translated into the target language. Then the translation probability is finally estimated considering the probability scores and reordering scores of the phrase pairs used in each step of derivation. PBMT follows the log-linear model of the Equation 2.6 to search for the translation candidate with the highest translation probability score, and the feature functions include both directions of phrase translation probability, both directions of lexical translation probability, word penalty, phrase penalty, and language model score.

PBMT is used in many research and practical systems because it can easily train and perform high-speed translation just by preparing a bilingual corpus

[2]In languages such as Japanese, Chinese, and Thai where words are not separated by spaces in ordinary texts, first it is necessary to perform tokenization using word segmentation tools.

between language pairs to be translated. However, since it is a method not considering the structure of sentences, it is difficult to effectively address the problem of word reordering. Although it is possible to introduce advanced word reordering models (Galley et al., 2004; Goto et al., 2013), reordering over long distances is still difficult and it is not easy to use in pivot translation.

2.1.4 Synchronous Context-Free Grammars

This section first covers Synchronous Context-Free Grammars (SCFGs), which are widely used in machine translation, particularly hierarchical phrase-based SMT (Hiero) (Chiang, 2007). In SCFGs, the elementary structures used in translation are synchronous rewrite rules with aligned pairs of source and target symbols on the right-hand side:

$$X \rightarrow \langle \overline{s},\ \overline{t} \rangle \tag{2.14}$$

where X is the head symbol of the rewrite rule, and $\overline{s}$ and $\overline{t}$ are both strings of terminals and non-terminals on the source and target side respectively. Each string in the right side pair has the same number of indexed non-terminals, and identically indexed non-terminals correspond to each-other. For example, a synchronous rule could take the form of:

$$X \rightarrow \left\langle X_0 \ \text{ of } \ X_1,\ X_1 \text{ 的 } X_0 \right\rangle. \tag{2.15}$$

Synchronous rules can be extracted based on parallel sentences and automatically obtained word alignments, similarly to PBMT. Each extracted rule is scored with phrase translation probabilities in both directions $\phi(\overline{s}|\overline{t})$ and $\phi(\overline{t}|\overline{s})$, lexical translation probabilities in both directions $\phi_{lex}(\overline{s}|\overline{t})$ and $\phi_{lex}(\overline{t}|\overline{s})$, a word penalty counting the terminals in $\overline{t}$, and a constant phrase penalty of 1.

At translation time, the decoder searches for the target sentence that maximizes the derivation probability, which is defined as the sum of the scores of the rules used in the derivation, and the log of the language model probability over the target strings. When not considering a language model, it is possible to efficiently find the best translation for an input sentence using the CKY+ algorithm

(Chappelier et al., 1998). When using a language model, the expanded search space is further reduced based on a limit on expanded edges, or total states per span, through a procedure such as cube pruning (Chiang, 2007).

2.1.5 Hierarchical Rules

In this section, we specifically cover the rules used in Hiero. Hierarchical rules are composed of initial head symbol S, and synchronous rules containing terminals and single kind of non-terminal X.[3] Hierarchical rules are extracted using the same phrase extraction procedure used in phrase-based translation (Koehn et al., 2003) based on word alignments, followed by a step that performs recursive extraction of hierarchical phrases (Chiang, 2007).

For example, hierarchical rules could take the form of:

$$X \rightarrow \left\langle \text{Officers, 主席团 成員} \right\rangle \tag{2.16}$$

$$X \rightarrow \left\langle \text{the Committee, 委员会} \right\rangle \tag{2.17}$$

$$X \rightarrow \left\langle X_0 \text{ of } X_1,\ X_1 \text{ 的 } X_0 \right\rangle. \tag{2.18}$$

From these rules, we can translate the input sentence by derivation:

$$\begin{aligned} S &\rightarrow \langle X_0,\ X_0 \rangle \\ &\Rightarrow \left\langle X_1 \text{ of } X_2,\ X_2 \text{ 的 } X_1 \right\rangle \\ &\Rightarrow \left\langle \text{Officers of } X_2,\ X_2 \text{ 主席团 成員} \right\rangle \\ &\Rightarrow \langle \text{Officers of the Committee,} \\ &\qquad \text{委员会 的 主席团 成員} \rangle. \end{aligned}$$

Hiero has an advantage that it is able to achieve relatively high word re-ordering accuracy (compared to other symbolic SMT alternatives such as standard PBMT) without language-dependent processing. On the other hand, since it does not use syntactic information and tries to extract all possible combinations of rules, it has the tendency to extract very large translation rule tables and also tends to be less syntactically faithful in its derivations.

[3] It is also standard to include glue rules $S \rightarrow \langle X_0,\ X_0 \rangle$, $S \rightarrow \langle S_0\ X_1,\ S_0\ X_1 \rangle$, $S \rightarrow \langle S_0\ X_1,\ X_1\ S_0 \rangle$ to fall back on when standard rules can not result in a proper derivation.

2.1.6 Explicitly Syntactic Rules

An alternative to Hiero rules is the use of synchronous context-free grammar or synchronous tree-substitution grammar (Graehl and Knight, 2004) rules that explicitly take into account the syntax of the source side (tree-to-string rules), target side (string-to-tree rules), or both (tree-to-tree rules). Taking the example of tree-to-string (T2S) rules, these use parse trees on the source language side, and the head symbols of the synchronous rules are not limited to S or X, but instead use non-terminal symbols corresponding to the phrase structure tags of a given parse tree. For example, T2S rules could take the form of:

$$X_{\text{NP}} \rightarrow \left\langle \text{(NP (NNS Officers))},\ \text{主席团 成員} \right\rangle, \tag{2.19}$$

$$X_{\text{NP}} \rightarrow \left\langle \text{(NP (DT the) (NNP Committee))},\ \text{委员会} \right\rangle, \tag{2.20}$$

$$X_{\text{PP}} \rightarrow \left\langle \text{(PP (IN of) } X_{\text{NP},0}\text{)},\ X_0\ \text{的} \right\rangle, \tag{2.21}$$

$$X_{\text{NP}} \rightarrow \left\langle \text{(NP } X_{\text{NP},0}\ X_{\text{PP},1}\text{)},\ X_1\ X_0 \right\rangle. \tag{2.22}$$

Here, parse subtrees of the source language rules are given in the form of S-expressions. From these rules, we can translate from the parse tree of the input sentence by derivation:

$$\begin{aligned}
X_{\text{ROOT}} &\rightarrow \left\langle X_{\text{NP},0},\ X_0 \right\rangle \\
&\Rightarrow \left\langle \text{(NP } X_{\text{NP},1}\ X_{\text{PP},2}\text{)},\ X_2\ X_1 \right\rangle \\
&\Rightarrow \left\langle \text{(NP (NP (NNS Officers) } X_{\text{PP},2}\text{))},\ X_2\ \text{主席团 成員} \right\rangle \\
&\overset{*}{\Rightarrow} \left\langle \begin{array}{l} \text{(NP} \\ \quad \text{(NP (NNS Officers))} \\ \quad \text{(PP (IN of)} \\ \quad\quad \text{(NP (DT the)} \\ \quad\quad\quad \text{(NNP Committee))))} \end{array} ,\ \text{委员会 的 主席团 成員} \right\rangle.
\end{aligned}$$

In this way, it is possible in T2S translation to obtain a result conforming to the source language's grammar. This method also has the advantage the number of less-useful synchronous rules extracted by syntax-agnostic methods such as Hiero are reduced, making it possible to train more compact rule tables and allowing for faster translation.

2.1.7 Multi-Synchronous Context-Free Grammars

Multi-Synchronous Context-Free Grammars (MSCFGs (Neubig et al., 2015)) are a generalization of SCFGs that are be able to generate sentences in multiple target languages simultaneously. The single target side string $\bar{t}$ in the SCFG production rule is extended to have strings for N target languages:

$$X \rightarrow \langle \bar{s},\ \overline{t_1},\ ...,\ \overline{t_N} \rangle\,. \tag{2.23}$$

Performing multi-target translation with MSCFGs is quite similar to translating using standard SCFGs, with the exception of the expanded state space caused by having one language model for each target. Neubig et al. (2015) has proposed a *sequential* search method, that ensures diversity in the primary target search space by first expanding with only primary target language model, then additionally expands the states for other language models, a strategy we also adopt in this work.

In the standard training method for MSCFGs, the multi-target rewrite rules are extracted from multilingual sentence-aligned corpora by applying an extended version of the standard SCFG rule extraction method, and scored with features that consider the multiple targets. It should be noted that this training method requires a large amount of sentence-aligned training data including the source and all target languages. This assumption breaks down when we have little parallel data, and thereby we propose a method to generate MSCFG rules by triangulating 2 SCFG rule tables in Chapter 4.

2.2 Neural Machine Translation

As shown previously, the training of SMT is carried out through various steps, and it is necessary to prepare independent translation knowledge such as phrase/rule table and language model, and design features to be considered at translation. On the other hand, in NMT, training and translation can all be done in the same framework by preparing only a single NN. Just by providing bilingual sentence pairs, NNs automatically learn some information necessary for translation. Such a

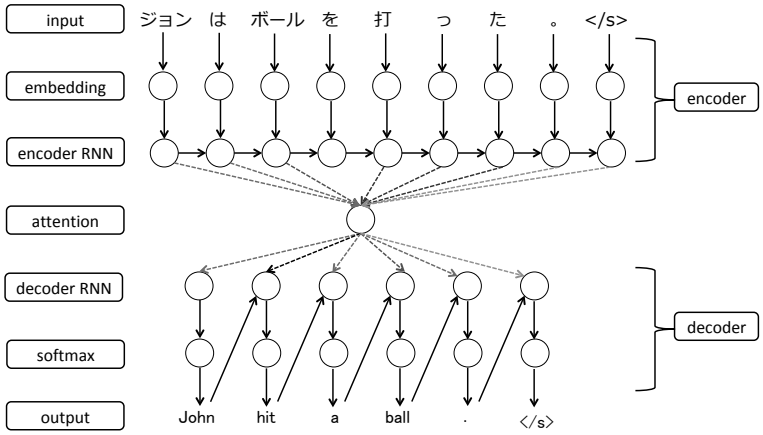


Figure 2.3: Attention-based neural MT

framework in which input to output is completed in a single model is called *end-to-end*. In this section, we introduce the commonly used elements in NMT: encoder-decoder model (Sutskever et al., 2014) and attention mechanism (Bahdanau et al., 2015; Luong et al., 2015).

NMT is largely divided into three parts. The first is an encoder that encodes an input sentence into a vector representation of continuous values. The second is an attention mechanism that controls where in the encoded input sentence should be noticed when determining the word to be output. The third is a decoder that decodes (generates) the output sentence based on the encoded input sentence and the attention information. A schematic diagram of NMT composed of these three parts is shown in Figure 2.3. In general, a model consisting of an encoder that encodes input information and a decoder that decodes the encoded information and obtains a desired output is called an encoder-decoder model. This model is applied in various tasks such as document summarization and image caption generation. Each element will be explained in the following subsections.

2.2.1 Encoder

The encoder first coverts each language into a vector of continuous values consisting hundreds of dimensions called *distributed representation*. This operation is called *embedding* (Mikolov et al., 2013). Although such distributed represen-

tation can be trained from a large-scale monolingual corpus independent from NMT, training of distributed representation is also commonly trained simultaneously as part of the network in NMT.

In embedding, the information of each word is represented independently of each other. Therefore, words whose meanings are determined by association with other words, such as function words and compound nouns, can not be represented well. To deal with this problem, the recurrent NN (RNN) layer reads each word one by one, and creates a vector representation that considers words before or after.

A basic NN only transmit information in one direction and do not have the function of using past information or storing information, though a RNN has a mechanism to feed back its own output as its input again. The hidden states of the forward RNN are computed as:

$$\overrightarrow{\boldsymbol{h}}_i = f\left(\overrightarrow{\boldsymbol{h}}_{i-1}, s\right) \tag{2.24}$$

where f is the RNN computation, $\overrightarrow{\boldsymbol{h}}_i$ is the hidden states of the forward RNN in time step t, and s is the source sentence representation. This mechanism makes it possible to use past information and current information at the same time. Since only the previous words can be considered in a forward RNN alone, we often use it as as a bidirectional RNN in combination with a backward RNN that allows for consideration of the following words:

$$\overleftarrow{\boldsymbol{h}}_i = f\left(\overleftarrow{\boldsymbol{h}}_{i+1}, s\right), \tag{2.25}$$

$$\overline{\boldsymbol{h}}_i = concat\left[\overrightarrow{\boldsymbol{h}}_i; \overleftarrow{\boldsymbol{h}}_i\right]. \tag{2.26}$$

2.2.2 Attention Mechanism

The attention mechanism plays a role of determining the point to be noticed when translating the next word with the encode input sentence and the internal state of the decoder as the decision information. Attention weights for each $\overline{\boldsymbol{h}}_i$ in each step t are computed as:

$$a_t(i) = \frac{score\left(\overline{\boldsymbol{h}}_i, \boldsymbol{h}_{t-1}\right)}{\sum_j score\left(\overline{\boldsymbol{h}}_j, \boldsymbol{h}_{t-1}\right)} \tag{2.27}$$

where the attentional weight $a_t(i)$ is determined by a score function, which receives one encoder state $\overline{\boldsymbol{h}}_i$ and the previous decoder state $\boldsymbol{h}_t$ as input. Every time one word is translated, these attention weights are recalculated, and the points to be noticed are changed every time.

For the decoder part, a context vector c_t, which is a weighted summarization of encoder states is calculated as:

$$c_t = \sum_i a_t(i)\overline{\boldsymbol{h}}_i. \tag{2.28}$$

2.2.3 Decoder

The decoder is generally composed of a single RNN, receives the context vector and the information of the word outputted one before, and outputs the next word. For this reason, translated sentences are generated one word at a time in order from the beginning. The output from RNN is a vector having the same number of dimensions as the number of unique words in the target language. The computation of each hidden state $\boldsymbol{h}_t$ of the decoder RNN is shown as follows:

$$\boldsymbol{h}_t = f\left(\boldsymbol{h}_{t-1}, c_t, y_{t-1}\right), \tag{2.29}$$

$$p(y_j|y_{<j}, \boldsymbol{s}) = softmax\left(\boldsymbol{W}_o\boldsymbol{h}_t + \boldsymbol{b}_o\right) \tag{2.30}$$

where y_{t-1} is the embedding of previous generated word, and directly drawn from the target translation. The softmax function normalizes the output of RNN layer like a probability distribution, and outputs the word with the highest probability. This is the outline of the mechanism of NMT.

3 Low-Resource Machine Translation

In Chapter 2, we mentioned that SMT can obtain translation rules automatically from a bilingual corpus, and searches for the translation candidate that maximizes the translation probability score. Since it is based on statistical models, the reliability of the probability estimation improves as the target-side monolingual corpus used for training the language model and the bilingual corpus used for training the translation model become large, and higher translation accuracy can be expected. Although there are influences such as the number of speakers or internet users in the target language, the language model is unlikely to be a problem because it is relatively easy to acquire training data. On the other hand, the bilingual corpus is the key issue to SMT, and it is impossible to translate words or expressions not covered by the training data. Therefore it is desirable to acquire as larger a bilingual corpus as possible. It has been reported that building a practical SMT system requires more than million sentence pairs (Dyer et al., 2008; Koehn et al., 2007). However, considering language pairs not including English, such as Japanese and French, while it is possible to acquire abundant monolingual corpora in each language, it is difficult to acquire a bilingual corpus of more than 1M sentence pairs. Thus, the bilingual corpus, which is a major premise of SMT, can not be immediately acquired in a sufficient size in many language pairs, and there is a problem in performing translation between an arbitrary language pair.

This chapter describes the two representative approaches for coping with the scarceness of bilingual corpus: pivot translation (Section 3.1) and active learning for SMT (Section 3.2).

3.1 Pivot Translation

Several methods have been proposed for SMT using pivot languages. These are categorized into 3 categories: sequential pivot translation (Section 3.1.1), pseudo-parallel corpus synthesis (Section 3.1.2), and triangulation of translation models (Section 3.1.3), these are covered in following sections.

3.1.1 Sequential Pivot Translation

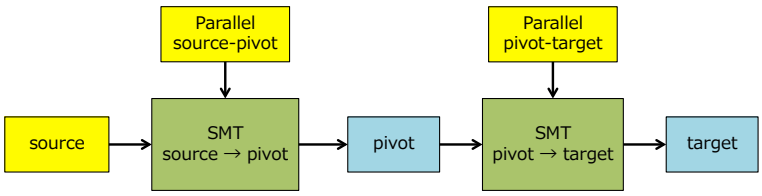


Figure 3.1: Sequential pivot translation

Figure 3.1 shows a diagram of translation from source sentence to target sentence by *sequential pivot translation* method (referred as *cascade* in experiments) (de Gispert and Mariño, 2006). This method first trains source-pivot and pivot-target MT models using respective bilingual corpora. Then, it is possible to translate from a source sentence into the target language by translating the source sentence into the pivot language and translating again the pivot sentence into the target language. Since this method pipelines only input and output of MT systems, it is not necessary in use only SMT and any MT systems can be combined. The advantage of this method is its ease of implementation just reusing available systems, and accurate pivot translation can be expected if two accurate MT systems are given. However, there are disadvantages that additive error propagated from two systems tends to damage translation accuracy, and optimization of the entire connected system is complicated.

In naïve implementation of this method, a source-pivot system may translate into only a single pivot sentence with the highest translation probability, though multi-sentential implementation has been proposed (Utiyama and Isahara, 2007) to expand the search space by leaving top n candidates with the highest translation probability as shown in Figure 3.2. However, n times more decoding time

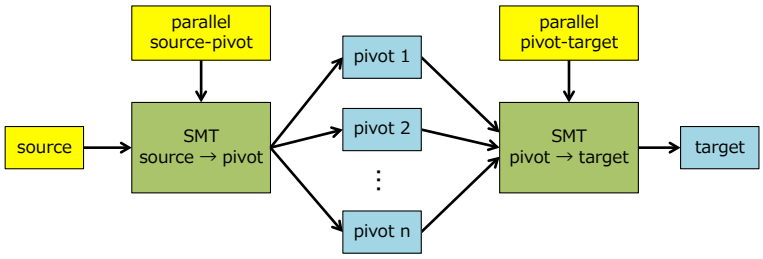


Figure 3.2: Multi-sentential method

is required than usual, and no significant improvement in accuracy has been reported (Utiyama and Isahara, 2007).

3.1.2 Pseudo-Parallel Corpus Synthesis

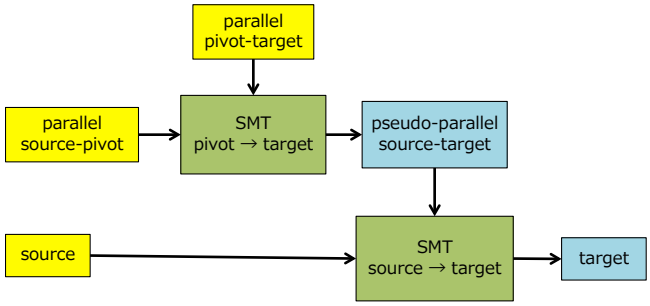


Figure 3.3: Pseudo-parallel corpus synthesis

Figure 3.3 shows a diagram of translation with SMT trained with source-target pseudo-parallel corpus by *pseudo-parallel corpus synthesis* method (referred as *synthetic* in experiments) (de Gispert and Mariño, 2006). This method first uses only one of source-pivot and pivot-target parallel corpora to train SMT system, for example, pivot-target is used in this figure. Then, source-target pseudo-parallel corpus is obtained by translating all the pivot-side sentences in source-pivot parallel corpus into target language. Thus, it is possible to train translation models using the obtained source-target pseudo-parallel corpus. Accurate translation can be expected if translation errors in the pseudo-parallel corpus do not significantly affect the training of statistical models. Since this method rebuilds new

training data from existing systems, it has the advantage that once it creates a pseudo-parallel corpus, it can use the same training procedure as regular SMT.

de Gispert and Mariño (2006) have performed experiments on pivot translation of Catalan and English using Spanish as a pivot language, comparing sequential pivot translation method and pseudo-parallel corpus synthesis. As a result, no significant difference between these methods has been reported.

3.1.3 Triangulation of Translation Models

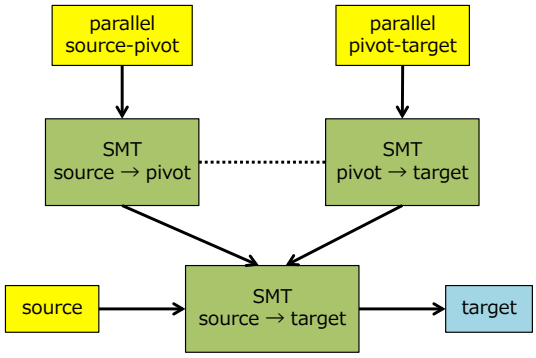


Figure 3.4: Triangulation of translation models

The training procedures of PBMT and SCFGs store the phrase pairs extracted and scored from a bilingual corpus into a structured file called the *phrase/rule table*. Figure 3.4 shows a diagram of *triangulation* of source-pivot and pivot-target phrase/rule tables into a source-target table (Cohn and Lapata, 2007; Utiyama and Isahara, 2007).

The triangulation method for PBMT first trains source-pivot and pivot-target phrase tables as T_{SP} and T_{PT} respectively. Then this method *triangulates* T_{SP} and T_{PT} by matching source-pivot and pivot-target phrase pairs having a common pivot phrase, and synthesizes them into source-target phrase pairs to create source-target phrase table T_{ST}. For all the combined source-target phrases, phrase translation probability $\phi(\cdot)$ and lexical translation probability $\phi_{lex}(\cdot)$ are estimated according to the following equations:

$$\phi(\bar{t}|\bar{s}) = \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi(\bar{t}|\overline{p}, \bar{s})\, \phi(\overline{p}|\bar{s}) \approx \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi(\bar{t}|\overline{p})\, \phi(\overline{p}|\bar{s}), \quad (3.1)$$

$$\phi(\bar{s}|\bar{t}) = \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi(\bar{s}|\overline{p}, \bar{t})\, \phi(\overline{p}|\bar{t}) \approx \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi(\bar{s}|\overline{p})\, \phi(\overline{p}|\bar{t}), \quad (3.2)$$

$$\phi_{lex}(\bar{t}|\bar{s}) = \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi_{lex}(\bar{t}|\overline{p}, \bar{s})\, \phi_{lex}(\overline{p}|\bar{s}) \approx \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi_{lex}(\bar{t}|\overline{p})\, \phi_{lex}(\overline{p}|\bar{s}), \quad (3.3)$$

$$\phi_{lex}(\bar{s}|\bar{t}) = \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi_{lex}(\bar{s}|\overline{p}, \bar{t})\, \phi_{lex}(\overline{p}|\bar{t}) \approx \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi_{lex}(\bar{s}|\overline{p})\, \phi_{lex}(\overline{p}|\bar{t}). \quad (3.4)$$

where $\bar{s}$, $\overline{p}$ and $\bar{t}$ are phrases in source, pivot and target respectively, and $\overline{p} \in T_{SP} \cap T_{PT}$ indicates that $\overline{p}$ is commonly contained in phrase tables T_{SP} and T_{PT}.

The triangulation method for SCFGs (Miura et al., 2015) can be done following the same idea for PBMT, using pre-trained source-pivot and pivot-target rule tables T_{SP} and T_{PT} respectively. Then this method searches T_{SP} and T_{PT} for source-pivot and pivot-target rules having a common pivot symbols, and synthesize them into source-target rules to create rule table T_{ST}:

$$\begin{aligned} &X \rightarrow \langle \bar{s},\ \bar{t} \rangle \in T_{ST} \\ &s.t.\ X \rightarrow \langle \bar{s}, \overline{p} \rangle \in T_{SP}\ \wedge\ X \rightarrow \langle \overline{p}, \bar{t} \rangle \in T_{PT}. \end{aligned} \quad (3.5)$$

Although SCFG triangulation is differ from PBMT triangulation in that $\bar{s}$, $\overline{p}$ and $\bar{t}$ are symbol strings which may contain terminals and non-terminals, its procedure of estimating translation probability scores is same with Equations 3.1-3.4. Word penalty and phrase penalty $X \rightarrow \langle \bar{s},\ \bar{t} \rangle$ are set as the same values of $X \rightarrow \langle \overline{p},\ \bar{t} \rangle$.

Utiyama and Isahara (2007) have performed experiments on pivot translation of several language pairs with English as a pivot language, comparing the sequential pivot translation method and the triangulation method. As a result, it has been reported that the triangulation method achieved higher BLEU score than simple sequential pivot translation with $n = 1$ and multi-sentential method with $n = 15$.

3.1.4 Problems of Pivot-Side Ambiguity

Although triangulation is known for achieving higher translation accuracy than other simple methods and has become a popular and standard work of pivot trans-

lation nowadays, there still remains the problem of ambiguity. This subsection describes reason causing the problem and examples.

In triangulation, Equations (3.1)-(3.4) are based on the memoryless channel model, which assumes:

$$\phi\left(\overline{t}|\overline{p},\overline{s}\right) = \phi\left(\overline{t}|\overline{p}\right), \tag{3.6}$$

$$\phi\left(\overline{s}|\overline{p},\overline{t}\right) = \phi\left(\overline{s}|\overline{p}\right). \tag{3.7}$$

For example, in Equation (3.6), it is assumed that the translation probability of target phrase given pivot and source phrases is never affected by the source phrase. However, it is easy to come up with examples where this assumption does not hold.

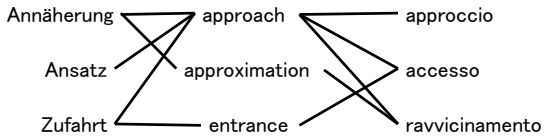


Figure 3.5: An example of ambiguity in De-En-It triangulation.

Figure 3.5 shows an example of three words in German and Italian each of which corresponds to the English polysemic word "approach." In such a case, finding associated source-target phrase pairs and estimating translation probabilities properly become complicated problems. As a result, pivot translation is significantly more ambiguous than standard translation.

This thesis distinguishes the ambiguity problem in pivot translation into two types: *semantic ambiguity* and *syntactic ambiguity*, and proposes approaches to resolve each of them.

3.1.5 Related Work

Up to this point, we have explained the representative pivot translation methods in SMT. Other related research in pivot translation is mainly based on the triangulation for PBMT, and focuses on discussion to further improve accuracy (Dabre et al., 2015; Levinboim and Chiang, 2015; Zhu et al., 2014). In the triangulation, it is a problem how to correctly estimate the translation probability.

Zhu et al. (2014) have proposed an estimation method of source-target translation probability by first estimating source-target co-occurrence counts instead of the direct estimation from source-pivot and pivot-target translation probabilities (Equations 3.1-3.4). They have reported that stable translation accuracy can be obtained even in triangulation of two phrase tables with unbalanced table size.

Levinboim and Chiang (2015) have reported that it is especially difficult to estimate word-level translation probability among phrase correspondence in the triangulation stage. Then they have proposed a method of improving the quality of the triangulation by estimating translation probability even for the correspondence of words which can not be directly observed, using distributed expression of words (Mikolov et al., 2013).

In this thesis, we discuss pivot translation with English as a pivot language, though it is also known that the translation accuracy varies depending on how to select a pivot language. The influence of pivot language selection on pivot translation has been discussed in detail by Paul et al. (2009). In reality, there are not many situations in which the pivot language can be selected from multiple candidates, though in the case where bilingual corpora of the same scale can be obtained via several languages, we should choose a pivot language having similar language structure with source or target language.

In addition, there is no need to limit the number of pivot languages necessarily to one, and methods to consider multiple pivot languages at the same time have been also proposed. As representative for that purpose, there are methods such as aggregating multiple source-target phrase/rule tables obtained by triangulation with respective pivot languages into one table with linear interpolation, and searching by considering multiple translation models simultaneously (Dabre et al., 2015).

Here we mention the relationship between pivot language in pivot translation and interlingual language (*interlingua* (Vauquois, 1968)) in RBMT. An attempt to MT using interlingua has a long history, and before the SMT was invented, the levels of translation has been discussed from the beginning when RBMT has been devised (Nirenburg, 1989). Figure 3.6 is called *Vauquois pyramid* and famous as a diagram showing the levels of analysis and generation in MT. It is expected that source language text is translated with less information loss by analyzing to

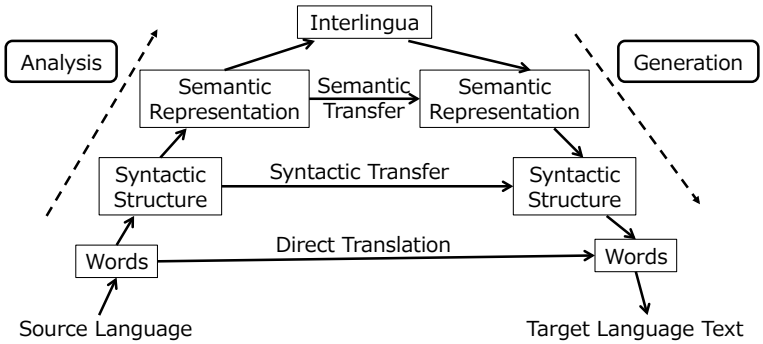


Figure 3.6: Levels of translation ("Vauquois pyramid")

higher level and transferring into target language. Interlingua, which is the top of the pyramid, is an ideal language that can cover expressions of all the languages, though such natural languages do not exist. In the past, there has been an attempt to convert arbitrary text into an artificial interlingua with RBMT which describes translation rules manually. However, with such rule-based approaches, it is impossible to cover various domains and expressions with hands of experts, and indeed, there is no human being who is familiar with every language.

With the invention of SMT, using bilingual corpora it became possible to translate highly accurately far more efficiently than rule description by experts. However, it is said that today's MT systems are wandering around the bottom of Vauquois pyramid. When an arbitrary natural language is used as a pivot language such as interlingua, some information might be lost depending on the expressiveness of the selected language, and then it may be impossible to reproduce the information of the source language text in the process of generation. For this reason, a form with higher expressiveness than an simple word sequence should be used for pivot language, and as an example, MT frameworks using ontology as intermediate representation have been proposed (Hovy, 1998). In addition, training methods of multilingual NMT, which improve translation accuracy by causing translation tasks of plural language pairs to be trained as a common encoder, have been also proposed (Dong et al., 2015; Johnson et al., 2017; Zoph and Knight, 2016). This can be interpreted as handling distributed representation of translation units well as an pivot language.

3.2 Active Learning

Active learning is a framework that makes it possible to efficiently train statistical models by selecting informative examples from a pool of unlabeled data. Most work on active learning for SMT, and natural language tasks in general, has focused on choosing which *sentences* to give to annotators. These methods generally assign priority to sentences that contain data that is potentially useful to the MT system according to a number of criteria.

For example, there are methods to select sentences that contain phrases that are frequent in monolingual data but not in bilingual data (Eck et al., 2005), have low confidence according to the MT system (Haffari et al., 2009), or are predicted to be poor translations by an MT quality estimation system (Ananthakrishnan et al., 2010a). However, while the selected sentences may contain useful phrases, they will also generally contain many already covered phrases that nonetheless cost time and money to translate.

To solve the problem of wastefulness in full-sentence annotation for active learning, there have been a number of methods proposed to perform *sub-sentential annotation of short phrases* for natural language tasks (Bloodgood and Callison-Burch, 2010; Settles and Craven, 2008; Sperber et al., 2014; Tomanek and Hahn, 2009). For MT in particular, Bloodgood and Callison-Burch (2010) have proposed a method that selects poorly covered n-grams to show to translators, allowing them to focus directly on poorly covered parts without including unnecessary words (Section 3.2.3).

3.2.1 Active Learning for Machine Translation

In this section, we first provide an outline of the active learning procedure to select phrases for SMT data. We regard a *segment* as a word sequence with arbitrary length, which indicates that full sentences and single words both qualify as segments. In Algorithm 1, we show the general procedure of incrementally selecting the next candidate for translation from the source language corpus, requesting and collecting the translation in the target language, and retraining the models.

In lines 1-4, we define the datasets and initialize them. *SrcPool* is a set with

Algorithm 1 Active learning for SMT

1: **Init:**
2: $SrcPool \leftarrow$ source language data including candidates for translation
3: $Translated \leftarrow$ translated parallel data
4: $Oracle \leftarrow$ oracle giving the correct translation for an input phrase
5: **Loop Until** *StopCondition*:
6: $TM \leftarrow TrainTranslationModel(Translated)$
7: $NewSrc \leftarrow SelectNextSegment(SrcPool, Translated, TM)$
8: $NewTrg \leftarrow GetTranslation(Oracle, NewSrc)$
9: $Translated \leftarrow Translated \cup \{\langle NewSrc, NewTrg \rangle\}$

each sentence in source language corpus as an element. $Translated$ indicates a set with source and target language phrase pairs. $Translated$ may be empty, but in most cases it consists of a seed corpus upon which we would like to improve. $Oracle$ is an oracle (e.g. a human translator), that we can query for a correct translation for an arbitrary input phrase.

In lines 5-9, we train models incrementally. *StopCondition* in line 5 is an arbitrary timing when to stop the loop, such as when we reach an accuracy goal or when we expend our translation budget. In line 6, we train the translation model using $Translated$, the available parallel data at this point. We evaluate the accuracy after training the translation model for each step in the experiments. In line 7, we select the next candidate for translation using features of $SrcPool$, $Translated$ and TM to make the decision.

In the following sections, we discuss existing methods to implement the selection criterion in line 7.

3.2.2 Sentence Selection using *N*-Gram Frequency

The first traditional method that we cover is a sentence selection method. Specifically, it selects the sentence including the most frequent uncovered phrase with a length of up to n words in the source language data. This method enables us to effectively cover the most frequent n-gram phrases and improve accuracy with fewer sentences than random selection. Bloodgood and Callison-Burch (2010) demonstrate results of a simulation showing that this method required less than

80% of the data required by randomly selected sentences to obtain the same accuracy.

However, the selected full sentences include many phrases already covered in the parallel data. This may cause an additional cost for words in redundant segments, a problem resolved by the phrase selection approach detailed in the following section.

3.2.3 Phrase Selection using *N*-Gram Frequency

In the second approach, we directly select and translate n-gram phrases that are the most frequent in the source language data but not yet covered in the translated data (Bloodgood and Callison-Burch, 2010). This method allows for improvement of coverage with fewer additional words than sentence selection, achieving higher efficiency by reducing the amount of data unnecessarily annotated. Bloodgood and Callison-Burch (2010) showed that by translating the phrases selected by this method using a crowdsourcing website, it was possible to achieve a large improvement of BLEU score, outperforming similar sentence-based methods.

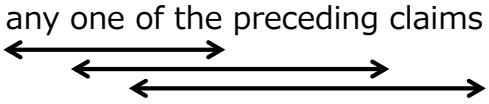


Figure 3.7: An example of n-gram selection method ($n = 4$)

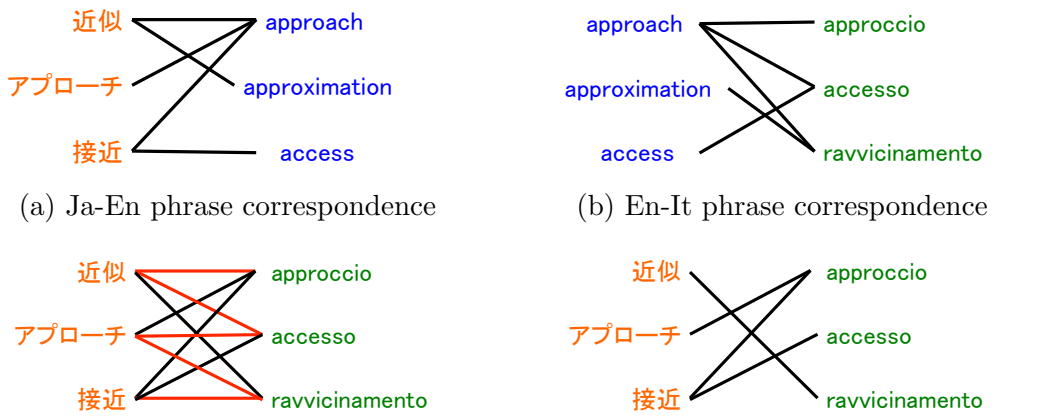
Nevertheless, our experiments identified two major practical problems with this method. First, as shown in Figure 3.7, many of the selected phrases overlap with each other, causing translation of *redundant* phrases, damaging efficiency. Second, it is common to see *fragments* of complex phrases such as "one of the preceding," which may be difficult for workers to translate into a contiguous phrase in the target language.

In addition to the two major issues, this method limits the maximum phrase length to $n = 4$, precluding the use of longer phrases. However, using a larger limit such as $n = 5$ is not likely to be a fundamental solution, as it increases the number of potentially overlapping phrases, and also computational burden.

4 Contextual Disambiguation in Pivot Translation

As mentioned in Section 3.1.3, triangulation method should estimate corresponding source-target phrase pairs given source-pivot and pivot-target phrase pairs and their probability scores. Section 3.1.4 mentioned that pivot-side ambiguity causes difficulty of triangulation. This chapter discusses the problem of semantic ambiguity due to word sense ambiguity and interlingual differences, and a method to solve this problem by incorporating pivot-side contextual information.

4.1 Word Sense Ambiguity



(a) Ja-En phrase correspondence

(b) En-It phrase correspondence

(c) All the Ja-It candidates

(d) Correct Ja-It phrase correspondence

Figure 4.1: Estimation of source-target word correspondence

This section illustrates the difficulty of triangulation due to polysemy. In Figure 4.1, it is shown that (a) Japanese-English and (b) English-Italic phrase cor-

respondence are given in individually trained translation models, then (c) many candidates to be triangulated resulting difficulty to accurately connect as (d) the correct Japanese-Italic phrase correspondence.

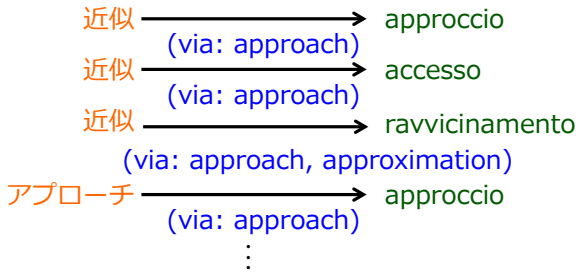


Figure 4.2: Standard triangulated phrases

Furthermore, in the conventional triangulation method, information about pivot phrases that behave as bridges between source and target phrases is lost after learning phrase pairs, as shown in Figure 4.2.

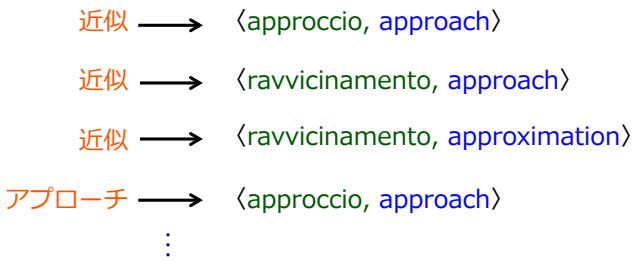


Figure 4.3: Proposed triangulated phrases

To overcome the problem, this thesis proposes a novel triangulation method that *remembers* the pivot phrase connecting source and target in the records of phrase/rule table, and estimates a joint translation probability from the source to target and pivot simultaneously. Figure 4.3 shows an example of triangulated phrases in this proposed method. The advantage of this approach is that generally we can obtain rich monolingual resources in pivot languages such as English, and SMT can utilize this additional information to improve the translation quality.

4.2 Triangulation Remembering the Pivot

To help reduce this ambiguity in standard triangulation method, our proposed triangulation method remembers the corresponding pivot phrase as additional information to be utilized for disambiguation. Specifically, instead of marginalizing over the pivot phrase $\overline{p}$ (Section 3.1.3), we create an MSCFG rule (Section 2.1.7) for the tuple of the connected source-target-pivot phrases such as:

$$\begin{aligned} & X \rightarrow \langle \overline{s}, \overline{t}, \overline{p} \rangle \in T_{STP} \\ & s.t.\ X \rightarrow \langle \overline{s}, \overline{p} \rangle \in T_{SP} \ \wedge \ X \rightarrow \langle \overline{p}, \overline{t} \rangle \in T_{PT} \end{aligned} \tag{4.1}$$

where T_{STP} indicates the MSCFG rule table triangulated from SCFG rule tables T_{SP} and T_{PT}. The advantage of translation with these rules is that they allow for incorporation of additional features over the pivot sentence such as a strong pivot language models.

In addition to the equations 3.1-3.4, we also estimate translation probabilities $\phi(\overline{t}, \overline{p}|\overline{s})$, $\phi(\overline{s}|\overline{p}, \overline{t})$ that consider both target and pivot phrases at the same time according to:

$$\phi\left(\overline{t}, \overline{p}|\overline{s}\right) = \phi\left(\overline{t}|\overline{p}\right)\phi\left(\overline{p}|\overline{s}\right), \tag{4.2}$$

$$\phi\left(\overline{s}|\overline{p}, \overline{t}\right) = \phi\left(\overline{s}|\overline{p}\right). \tag{4.3}$$

Translation probabilities between source and pivot phrases $\phi(\overline{p}|\overline{s})$, $\phi(\overline{s}|\overline{p})$, $\phi_{lex}(\overline{p}|\overline{s})$, and $\phi_{lex}(\overline{s}|\overline{p})$ can also be used directly from the source-pivot rule table. This results in 13 features for each MSCFG rule: 10 translation probabilities $\phi\left(\overline{t}|\overline{s}\right)$, $\phi\left(\overline{s}|\overline{t}\right)$, $\phi\left(\overline{p}|\overline{s}\right)$, $\phi\left(\overline{s}|\overline{p}\right)$, $\phi_{lex}\left(\overline{t}|\overline{s}\right)$, $\phi_{lex}\left(\overline{s}|\overline{t}\right)$, $\phi_{lex}\left(\overline{p}|\overline{s}\right)$, $\phi_{lex}\left(\overline{s}|\overline{p}\right)$, $\phi\left(\overline{t}, \overline{p}|\overline{s}\right)$, $\phi\left(\overline{s}|\overline{p}, \overline{t}\right)$, 2 word penalties counting the terminals in $\overline{t}$ and $\overline{p}$, and a constant phrase penalty of 1.

It should be noted that remembering the pivot results in significantly larger rule tables. To save computational resources, several pruning methods are conceivable. Neubig et al. (2015) show that an effective pruning method in the case of a main target T_1 with the help of target T_2 is the T_1-pruning method, namely, using L candidates of $\overline{t_1}$ with the highest translation probability $\phi(\overline{t_1}|\overline{s})$ and selecting $\overline{t_2}$ with highest $\phi(\overline{t_1}, \overline{t_2}|\overline{s})$ for each $\overline{t_1}$. We follow this approach, using the L best $\overline{t}$, and the corresponding 1 best $\overline{p}$.

4.3 Experiments

To investigate the effect of the proposed approach, we perform experiments comparing pivot translation among multiple language pairs with the following procedure.

4.3.1 Experimental Set-Up

We evaluate the proposed triangulation method through pivot translation experiments on the United Nations Parallel Corpus (UN6Way (Ziemski et al., 2016)) corpus. UN6Way is a line-aligned multilingual parallel corpus that includes data in English (En), Arabic (Ar), Spanish (Es), French (Fr), Russian (Ru) and Chinese (Zh), covering different families of languages. It contains more than 11M sentences for each language pair, and is therefore suitable for multilingual translation tasks such as pivot translation. In these experiments, we fixed English as the pivot language considering that it is the language most frequently used as a pivot language. This has the positive side-effect that accurate phrase structure parsers are available in the pivot language, which is good for our proposed method. We perform pivot translation on all the combinations of the other 5 languages, and compared the accuracy of each method.

In pivot translation, source-pivot and pivot-target parallel corpus is respectively used to train translation models. However, it is assumed that direct sentence tuples of source, pivot, and target are rarely found in scenes where pivot translation is required. Therefore, we ensure that there will be no common pivot sentence in source-pivot and pivot-target corpus used for training in this experiment.

In our basic training setup, we use 100k sentences for training both the TMs and the target LMs. We assume that in many situations, a large amount of English monolingual data is readily available and therefore, we train pivot LMs with different data sizes up to 5M sentences. The archive of this corpus contains already split data sets: about 11M sentences for training, 4,000 sentences respectively for evaluation and for parameter tuning. As preprocessing, the parallel sentences including duplicated English sentence are removed to ensure uniqueness of pivot sentences, and the sentences of length over 60 words in training data are filtered out to ensure accuracy of word alignment, and the sentences

over 80 words in evaluation and tuning data are filtered out as well. Then, there remain about 8M sentences for training, and about 3,800 sentences respectively for evaluation and tuning. Since the plain texts of Chinese are not divided for words, they are tokenized with Chinese segmentation model of KyTea (Neubig et al., 2011). However, since the number of combinations to be evaluated is enormous, we hold out much small data sets compared with preprocessed data size: 100k sentences respectively for training source-pivot (referred as *train1*) and pivot-target (referred as *train2*) translation models, and 1,500 sentences respectively for evaluation and tuning, such that $train1$ and $train2$ do not have any duplicated English sentences.

As a decoder, we use Travatar (Neubig, 2013), and train SCFG TMs with its Hiero extraction code. All the translation tasks use 5-gram language models trained from 200k target-side sentences of $train1$+$train2$ using KenLM (Heafield, 2011) to evaluate naturalness during decoding. Translation results are evaluated by BLEU (Papineni et al., 2002) and we tuned to maximize BLEU scores using MERT (Och, 2003). For trained and triangulated TMs, we use T_1 rule pruning with a limit of 20 rules per source rule. For decoding using MSCFG, we adopt the sequential search method.

We evaluate 3 pivot translation methods and 1 direct translation method:

Cascade:
: Sequential pivot translation with source-pivot and pivot-target SCFG models (Section 3.1.1). "w/ PvtLM 200k/5M" indicates translation with pivot LM trained using 200k/5M sentences respectively.

Tri. SCFG:
: Triangulating source-pivot and pivot-target SCFG TMs into a source-target SCFG TM using the traditional method (Section 3.1.3).

Tri. MSCFG:
: Triangulating source-pivot and pivot-target SCFG TMs into a source-target-pivot MSCFG TM in our approach. "w/o PvtLM" indicates translating without a pivot LM and "w/ PvtLM 200k/5M" indicates a pivot LM trained using 200k/5M sentences respectively (proposed, Section 4.2).

Src	Trg	BLEU Score [%]						
		Direct 1/2	Cascade w/ PvtLM		Tri. SCFG	Tri. MSCFG		
						w/o PvtLM	w/ PvtLM	
			200k	5M	(baseline)		200k	5M
Ar	Es	29.12 / 29.41	26.78	‡ **28.52**	27.84	‡ **28.44**	‡ **28.48**	‡ **29.13**
	Fr	23.68 / 22.19	20.76	‡ **22.02**	21.37	‡ **21.94**	‡ **22.01**	‡ **22.52**
	Ru	17.28 / 16.82	14.71	15.91	16.22	**16.38**	† **16.61**	‡ **16.76**
	Zh	14.51 / 14.54	13.85	‡ **15.01**	14.38	**14.43**	‡ **14.93**	‡ **15.50**
Es	Ar	14.46 / 14.17	13.87	† **14.35**	13.97	**14.19**	† **14.34**	‡ **14.42**
	Fr	35.87 / 34.81	29.72	30.20	32.62	**32.70**	**32.81**	† **32.94**
	Ru	21.58 / 22.18	19.55	20.52	20.91	**20.92**	**20.95**	‡ **21.52**
	Zh	17.56 / 18.10	17.54	‡ **18.37**	17.79	17.56	† **18.13**	‡ **18.70**
Fr	Ar	12.37 / 12.49	11.82	11.96	12.35	12.00	12.35	‡ **12.71**
	Es	38.34 / 38.82	33.16	33.90	36.10	36.00	**36.33**	‡ **36.93**
	Ru	20.10 / 20.86	18.33	18.95	19.51	† **19.91**	‡ **20.23**	‡ **20.44**
	Zh	15.99 / 15.90	15.73	**16.35**	16.17	**16.20**	**16.40**	† **16.58**
Ru	Ar	12.29 / 12.13	11.47	11.72	11.75	**11.84**	**11.91**	‡ **12.18**
	Es	31.01 / 31.39	28.67	‡ **30.23**	29.60	† **29.90**	‡ **30.59**	‡ **31.40**
	Fr	25.74 / 25.71	23.64	† **25.09**	24.60	**24.63**	‡ **25.20**	‡ **25.32**
	Zh	15.85 / 15.75	15.99	† **16.68**	16.12	15.75	‡ **16.66**	‡ **16.81**
Zh	Ar	10.25 / 10.13	9.85	9.79	9.90	**9.98**	**10.04**	**10.07**
	Es	23.55 / 23.76	23.10	† **23.98**	23.41	**23.62**	† **23.83**	‡ **24.45**
	Fr	18.83 / 18.72	18.15	**18.45**	18.39	18.30	**18.51**	‡ **19.16**
	Ru	14.96 / 15.44	13.92	13.65	14.00	**14.24**	‡ **14.41**	‡ **14.67**

Table 4.1: Results of each method, comparing the proposed triangulation remembering the pivot with other methods.

Direct:

Translating with a direct SCFG trained on the source-target parallel corpus (not using a pivot language) for comparison. "Direct 1/2" indicates the evaluation scores of SCFG models trained respectively with $train1/train2$.

4.3.2 Results and Analysis

The result of experiments using all combinations of pivot translation tasks via English is shown in Table 4.1. Significant difference test is tested for the results according to bootstrap resampling method (Koehn, 2004). Bold face indicates

higher BLEU score than the baseline triangulation Tri. SCFG, underline indicates the highest BLEU score in pivot translation, and daggers indicate statistically significant gains over Tri. SCFG ($\dagger : p < 0.05, \ddagger : p < 0.01$). From the results, we can see that the proposed triangulation method considering pivot language models outperforms the traditional triangulation method for all language pairs, and translation with larger pivot language models improves the BLEU scores. For all languages, the pivot-remembering triangulation method with the pivot language model trained with 5M sentences achieves the highest score of the pivot translation methods, with gains of up to 1.8 BLEU points from the baseline method. The average gain of BLEU score in Tri. MSCFG w/ PvtLM 5M is about 0.75 points. This shows that remembering the pivot and using it to disambiguate results is consistently effective in improving translation accuracy.

Also to investigate the effects of different factors separately, we compare triangulation method that creates MSCFG rule table but not uses pivot language model for decoding (Tri. MSCFG w/o PvtLM). We can also see that the MSCFG triangulated model without using the pivot language model slightly outperforms the standard SCFG triangulation method for the majority of language pairs. It is conceivable that the additional scores of translation probabilities with pivot phrases are effective features that allow for more accurate rule selection.

Since a method using a large-scale pivot language model is possible not only by proposed triangulation into MSCFG rule table, but also by conventional method, we evaluate the translation score of sequential pivot translation with pivot language model trained with 5M sentences (Cascade w/ PvtLM 5M). Cascade w/ PvtLM 5M achieves higher score in all language pairs except Zh-Ar and Zh-Ru, compared with the sequential pivot translation method which uses pivot language model of only 200k sentences, then it has been confirmed that using simply large scale of language model lead to improvement. However, this accuracy improvement depends on language pairs, and Cascade w/ PvtLM 5M did not result in stable improvement when compared with the conventional triangulation method Tri. SCFG. In addition, when comparing the proposed method Tri. MSCFG w/ PvtLM 5M with Cascade w/ PvtLM 5M, the proposed method achieves higher score in the tested language pairs in spite of using the same pivot language model, and thus it can be said that the combination of triangulation and large-scale pivot

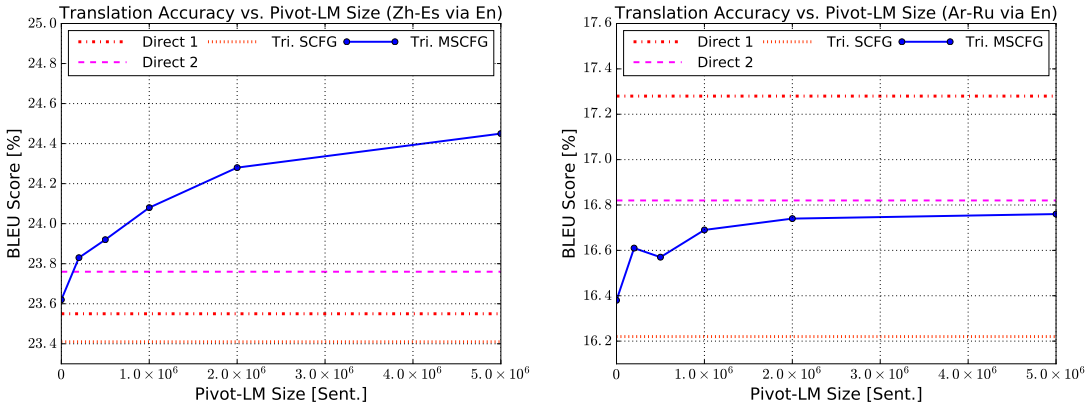


Figure 4.4: Influence of pivot LM size on pivot translation accuracy

language model is effective.

4.3.3 Influence of Pivot Language Model Strength

Although the magnitude of the influence of the pivot language model size on pivot translation differs on language pairs, it can also be confirmed that the accuracy improves as the training data size of the pivot language model increases. Figure 4.4 shows the influence the intermediate language model trained with different data sizes on the accuracy of Zh-Es (left) and Ar-Ru (right) translation. It can also be observed from the figure that the pivot language model helps to disambiguate and contributes to the improvement of translation accuracy. Although it can also be seen that the influence on the translation accuracy by increasing the training data size of pivot language model is logarithmic, this is the same tendency as the influence of the target language model size on the translation accuracy (Brants et al., 2007). In the Zh-Es translation, there is a prospect of accuracy improvement by using a larger pivot language model from the tendency of the graph, whereas in the case of Ar-Ru, there is a limit to further improvement since pivot language model of 5M sentences hardly improve over 2M sentences.

4.3.4 Qualitative Analysis

We show an example of a translated sentence for which pivot-side ambiguity seems to be resolved in the proposed triangulation method.

Input (French):

Le nom du candidat **proposé** est indiqué dans l'annexa à la présente note .

Reference (Spanish):

El nombre del candidato **propuesto** se presenta en el anexo de la presente nota .

Corresponding English Sentence:

The name of the candidate **thus nominated** is set out in the annex to the present note .

Tri. SCFG:

El nombre del **proyecto** de un candidato se indica en el anexo a la presente nota . (BLEU+1: 34.99)

Tri. MSCFG w/ PvtLM 5M:

El nombre del candidato **propuesto** se indica en el anexo a la presente nota . (BLEU+1: 61.13)

The name of the candidate **proposed** indicated in the annex to the present note . (Generated English Sentence)

In the example above, the French participle "prposé" (meaning "nominated") in the input sentence corresponds to the Spanish participle "propuesto" in translation. However, in the conventional triangulation method, the Spanish noun "proyecto" (meaning "project" or "plan"), inappropriate correspondence, is selected in derivation, causing incorrect translation. On the other hand, it seems that the translation result improved since "propuesto" in Spanish and "proposed" in English are simultaneously associated with "proposé" in the input sentence with the proposed method, prompting appropriate vocabulary selection from the context of the words in the generated English sentence.

Conversely, we show an example of translation result with no improvement, since proposed method does not work well.

Input (French):

J . Risques **d'aspiration** : **citère** de viscosité pour la classification des **mélanges** ;

Reference (Spanish):

J . Peligros por **aspiración** : **criterio** de viscosidad para la clasificación de **mezclas** ;

Corresponding English Sentence:

J . **Aspiration** hazards : viscosity **criterion** for classification of **mixtures** ;

Direct 1:

J . Riesgos **d'aspiration** : **criterio** de viscosité para la clasificación de **los mélanges** ; (BLEU+1: 34.20)

Direct 2:

J . Riesgos **d'aspiration** : **criterio** de viscosité para la clasificación de **mezclas** ; (BLEU+1: 49.16)

Tri. MSCFG w/ PvtLM 5M:

J . Riesgos **d'aspiration** : viscosité **criterios** para la clasificación de **mélanges** ; (BLEU+1: 27.61)

J . Risk d'aspiration : viscosité **criteria** for the categorization of **mélanges** ; (Generated English Sentence)

In this example, French technical terms such as "d'aspiration" (meaning "aspiration") and "mélanges" (meaning "mixture") are less frequent in the training corpus and, "d'aspiration" does not appear in both of *train1* and *train2*, thereof it is not translatable in both Direct 1/2 and treated as an unknown word, and "mélanges" is treated as unknown in Direct 1 since it appears only in *train2*. Since this problem can not be solved also by sequential pivot translation or pseudo-parallel corpus synthesis, it is necessary to supplement such uncovered expressions using external dictionaries or active learning methods.

Moreover in this example, besides the problem of unknown words, there is also a problem that the singular form of Spanish noun “criterio” (meaning “criterion”) is in plural form “criterios” in the proposed triangulation method, and word order is also wrong.

4.4 Summary

In this chapter, we have proposed a method for pivot translation using triangulation of SCFG rule tables into an MSCFG rule table that remembers the pivot, and performing translation with pivot LMs. In experiments, we found that these models are effective in the case when a strong pivot language model exists.

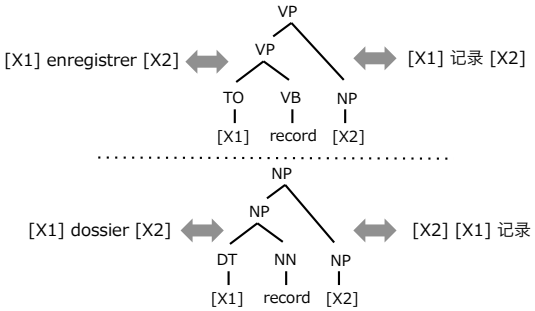
This proposed method is effective in the case that available source-pivot and pivot-target parallel corpora are not large (e.g. of less than hundred thousands sentence pairs) and amount of available pivot monolingual corpus is large on the contrary. Since MSCFG decoder demands huge amount of memory and computational time and is difficult to distributed processing, it is not realistic to scale up with the same framework. Moreover, although MSCFG decoder helps to select appropriate translation rule with pivot language model, it can not essentially solve the ambiguity problem and inappropriately connected rules still remain in the triangulated rule table as a noise. Therefore, in the following chapter, we discuss how to reduced the noisy rules in triangulated TMs and make it closer to directly trained TMs.

5 Syntactic Disambiguation in Pivot Translation

In Section 4, we showed example of semantic ambiguity and proposed triangulation method to consider pivot-side contextual information. However, there occurs not only semantic ambiguity but also syntactic ambiguity in pivot translation. This chapter discusses the problem of syntactic ambiguity due to the fact that parts-of-speech or phrase structures are not uniquely determined in local context.

5.1 Syntactic Ambiguity

(a) Standard triangulation method matching phrases



(b) Proposed triangulation method matching subtrees

Figure 5.1: Example of disambiguation by parse subtree matching (Fr-En-Zh)

In Figure 5.1 (a), we show an example of standard triangulation on Hiero TMs that combines hierarchical rules of phrase pairs by matching pivot phrases with equivalent surface forms. This example also demonstrates problems of ambiguity: the English word "record" can correspond to several different parts-of-speech according to the context. More broadly, phrases including this word also have different possible grammatical structures, but it is impossible to uniquely identify this structure unless information about the surrounding context is given.

This varying syntactic structure will affect translation. For example, the French verb "enregistrer" corresponds to the English verb "record", but the French noun "dossier" also corresponds to "record" — as a noun. As a more extreme example, Chinese is a languages that does not have inflections according to the part-of-speech of the word. As a result, even in the contexts where "record" is used with different parts-of-speech, the Chinese word ''记录" will be used, although the word order will change. These facts might result in an incorrect connection of "[X1] enregistrer [X2]" and "[X2] [X1] 记录" even though proper correspondence of "[X1] enregistrer [X2]" and "[X1] dossier [X2]" would be "[X1] 记录 [X2]" and "[X2] [X1] 记录". Hence a superficial phrase matching method based solely on the surface form of the pivot will often combine incorrect phrase pairs, causing translation errors if their translation scores are estimated to be higher than the proper correspondences.

Given this background, we hypothesize that disambiguation of these cases would be easier if the necessary syntactic information such as phrase structures are considered during pivoting. To incorporate this intuition into our models, we propose a method that considers syntactic information of the pivot phrase, as shown in Figure 5.1 (b). In this way, the model will distinguish translation rules extracted in contexts in which the English symbol string "[X1] record [X2]" behaves as a verbal phrase, from contexts in which the same string acts as a noun phrase.

While standard triangulation uses only the surface forms for performing triangulation, we propose two methods for triangulation based on syntactic matching (Section 5.2.1). The first places a hard restriction on exact matching of parse trees (Section 5.2.1) included in translation rules, while the second places a softer restriction allowing partial matches (Section 5.2.2).

5.2 Triangulation with Syntactic Information

In Section 5.1, we explained the standard triangulation method and mentioned that the pivot-side ambiguity causes incorrect estimation of translation probability and the translation accuracy might decrease. To address this problem, it is desirable to be able to distinguish pivot-side phrases that have different syntactic roles or meanings, even if the symbol strings are identical. In the following two sections, we describe two methods to distinguish pivot phrases that have syntactically different roles, one based on exact matching of parse trees, and one based on soft matching.

5.2.1 Exact Matching of Parse Subtrees

In the exact matching method, we first train pivot-source and pivot-target T2S TMs by parsing the pivot side of parallel corpora, and store them into rule tables as T_{PS} and T_{PT} respectively. Synchronous rules of T_{PS} and T_{PT} take the form of $X \rightarrow \langle \hat{p}, \overline{s} \rangle$ and $X \rightarrow \langle \hat{p}, \overline{t} \rangle$ respectively, where $\hat{p}$ is a symbol string that expresses pivot-side parse subtree (S-expression), $\overline{s}$ and $\overline{t}$ express source and target symbol strings. The procedure of synthesizing source-target synchronous rules essentially follows equations (3.1)-(3.4), except using T_{PS} instead of T_{SP} (direction of probability features is reversed) and pivot subtree $\hat{p}$ instead of pivot phrase $\overline{p}$. Here $\overline{s}$ and $\overline{t}$ do not have syntactic information, and thus the synthesized synchronous rules should be hierarchical rules explained in Section 2.1.5.

The matching condition of this method has harder constraints than matching of superficial symbols in standard triangulation, and has the potential to reduce incorrect connections of phrase pairs, resulting in a more reliable triangulated TM. On the other hand, the number of connected rules decreases as well in this restricted triangulation, and the coverage of the triangulated model might be reduced. Therefore it is important to create TMs that are both reliable and have high coverage.

5.2.2 Partial Matching of Parse Subtrees

To prevent the problem of the reduction of coverage in the exact matching method, we propose a partial matching method that keeps coverage just like standard triangulation by allowing connection of incompletely equivalent pivot subtrees. To estimate translation probabilities in partial matching, we first define *weighted triangulation* generalizing the equations (3.1)-(3.4) of standard triangulation with the weight function $\psi(\cdot)$:

$$\phi\left(\overline{t}|\overline{s}\right) = \sum_{\hat{p_T}} \sum_{\hat{p_S}} \phi\left(\overline{t}|\hat{p_T}\right) \psi\left(\hat{p_T}|\hat{p_S}\right) \phi\left(\hat{p_S}|\overline{s}\right), \tag{5.1}$$

$$\phi\left(\overline{s}|\overline{t}\right) = \sum_{\hat{p_S}} \sum_{\hat{p_T}} \phi\left(\overline{s}|\hat{p_S}\right) \psi\left(\hat{p_S}|\hat{p_T}\right) \phi\left(\hat{p_T}|\overline{t}\right), \tag{5.2}$$

$$\phi_{lex}\left(\overline{t}|\overline{s}\right) = \sum_{\hat{p_T}} \sum_{\hat{p_S}} \phi_{lex}\left(\overline{t}|\hat{p_T}\right) \psi\left(\hat{p_T}|\hat{p_S}\right) \phi_{lex}\left(\hat{p_S}|\overline{s}\right), \tag{5.3}$$

$$\phi_{lex}\left(\overline{s}|\overline{t}\right) = \sum_{\hat{p_S}} \sum_{\hat{p_T}} \phi_{lex}\left(\overline{s}|\hat{p_S}\right) \psi\left(\hat{p_S}|\hat{p_T}\right) \phi_{lex}\left(\hat{p_T}|\overline{t}\right) \tag{5.4}$$

where $\hat{p_S} \in T_{SP}$ and $\hat{p_T} \in P_{PT}$ are pivot parse subtrees of source-pivot and pivot-target synchronous rules respectively. By adjusting $\psi(\cdot)$, we can control the magnitude of the penalty for the case of incompletely matched connections. If we define $\psi(\hat{p_T}|\hat{p_S}) = 1$ when $\hat{p_T}$ is equal to $\hat{p_S}$ and $\psi(\hat{p_T}|\hat{p_S}) = 0$ otherwise, equations (5.1)-(5.4) are equivalent with equations (3.1)-(3.4).

Better estimating $\psi(\cdot)$ is not trivial, and co-occurrence counts of $\hat{p_S}$ and $\hat{p_T}$ are not available. Therefore we introduce a heuristic estimation method as follows:

$$\psi(\hat{p_T}|\hat{p_S}) = \frac{w(\hat{p_S}, \hat{p_T})}{\sum_{\hat{p} \in T_{PT}} w(\hat{p_S}, \hat{p})} \cdot \max_{\hat{p} \in T_{PT}} w(\hat{p_S}, \hat{p}) \tag{5.5}$$

$$\psi(\hat{p_S}|\hat{p_T}) = \frac{w(\hat{p_S}, \hat{p_T})}{\sum_{\hat{p} \in T_{SP}} w(\hat{p}, \hat{p_T})} \cdot \max_{\hat{p} \in T_{SP}} w(\hat{p}, \hat{p_T}) \tag{5.6}$$

$$w(\hat{p_S}, \hat{p_T}) = \begin{cases} 0 & (flat(\hat{p_S}) \neq flat(\hat{p_T})) \\ \exp\left(-d\left(\hat{p_S}, \hat{p_T}\right)\right) & (otherwise) \end{cases} \tag{5.7}$$

$$d(\hat{p_S}, \hat{p_T}) = TreeEditDistance(\hat{p_S}, \hat{p_T}) \tag{5.8}$$

where $flat(\hat{p})$ returns the symbol string of $\hat{p}$ keeping non-terminals, and $TreeEditDistance(\hat{p_S}, \hat{p_T})$ is minimum cost of a sequence of operations (contract

an edge, uncontract an edge, modify the label of an edge) needed to transform $\hat{p_S}$ into $\hat{p_T}$ (Klein, 1998).

According to equations (5.5)-(5.8), we can assure that incomplete match of pivot subtrees leads $d(\cdot) \geq 1$ and penalizes such that $\psi(\cdot) \leq 1/e^d \leq 1/e$, while exact match of subtrees leads to a value of $\psi(\cdot)$ at least $e \approx 2.718$ times larger than when using partially matched subtrees.

5.3 Experiments

5.3.1 Experimental Set-Up

To investigate the effect of our proposed approach, we evaluate the translation accuracy through pivot translation experiments on the United Nations Parallel Corpus (UN6Way) (Ziemski et al., 2016). UN6Way is a line-aligned multilingual parallel corpus that includes data in English (En), Arabic (Ar), Spanish (Es), French (Fr), Russian (Ru) and Chinese (Zh), covering different families of languages. It contains more than 11M sentences for each language pair, and is therefore suitable for multilingual translation tasks such as pivot translation. In these experiments, we fixed English as the pivot language considering that it is the language most frequently used as a pivot language. This has the positive side-effect that accurate phrase structure parsers are available in the pivot language, which is good for our proposed method. We perform pivot translation on all the combinations of the other 5 languages, and compared the accuracy of each method. For tokenization, we adopt SentencePiece,[1] an unsupervised text tokenizer and detokenizer, that is although designed mainly for neural MT, we confirmed that it also helps to reduce training time and even improves translation accuracy in our Hiero model as well. We first trained a single shared tokenization model by feeding a total of 10M sentences from the data of all the 6 languages, set the maximum shared vocabulary size to be 16k, and tokenized all available text with the trained model. We used English raw text without SentencePiece tokenization for phrase structure analysis and for training Hiero and T2S TMs on the pivot side. To generate parse trees, we used the Ckylark PCFG-LA parser (Oda

[1] https://github.com/google/sentencepiece

et al., 2015), and filtered out lines of length over 60 tokens from all the parallel data to ensure accuracy of parsing and alignment. About 7.6M lines remained. Since Hiero requires a large amount of computational resources for training and decoding, so we decided not to use all available training data but first 1M lines for training each TM. As a decoder, we use Travatar (Neubig, 2013), and train Hiero and T2S TMs with its rule extraction code. We train 5-gram LMs over the target side of the same parallel data used for training TMs using KenLM (Heafield, 2011). For testing and parameter tuning, we used the first 1,000 lines of the 4,000 lines test and dev sets respectively. For the evaluation of translation results, we first detokenize with the SentencePiece model and re-tokenized with the tokenizer of the Moses toolkit (Koehn et al., 2007) for Arabic, Spanish, French and Russian and re-tokenized Chinese text with KyTea tokenizer (Neubig et al., 2011), then evaluated using case-sensitive BLEU-4 (Papineni et al., 2002).

We evaluate 6 translation methods:

Cascade:
: Sequential pivot translation with source-pivot and pivot-target Hiero TMs (baseline, Section 3.1.1).

Tri. Hiero:
: Triangulating source-pivot and pivot-target Hiero TMs into a source-target Hiero TM using the traditional method (baseline, Section 3.1.3).

Tri. TreeExact
: Triangulating pivot-source and pivot-target T2S TMs into a source-target Hiero TM using the proposed exact matching of pivot subtrees (proposed 1, Section 5.2.1).

Tri. TreePartial
: Triangulating pivot-source and pivot-target T2S TMs into a source-target Hiero TM using the proposed partial matching of pivot subtrees (proposed 2, Section 5.2.2).

Direct:
: Translating with a Hiero TM directly trained on the source-target parallel corpus without using pivot language (as an oracle).

Source	Target	BLEU Score [%]				
		Direct	Cascade (baseline)	Tri. Hiero (baseline)	Tri. TreeExact (proposed 1)	Tri. TreePartial (proposed 2)
Ar	Es	38.49	30.95	34.20	‡ 34.97	‡ **35.94**
	Fr	33.34	25.08	29.93	‡ 30.68	‡ **30.83**
	Ru	24.63	18.70	22.94	‡ 23.94	‡ **24.15**
	Zh	27.27	21.77	22.78	‡ **25.17**	‡ 25.07
Es	Ar	27.18	22.72	22.97	‡ 24.09	‡ **24.45**
	Fr	43.24	35.40	38.74	‡ 39.62	‡ **40.12**
	Ru	28.83	22.43	26.35	‡ 27.25	‡ **27.41**
	Zh	27.08	23.36	24.54	25.00	† **25.16**
Fr	Ar	25.10	19.88	21.65	21.40	† **22.13**
	Es	45.20	37.75	40.16	‡ 41.03	‡ **41.99**
	Ru	27.42	20.64	24.71	† 25.24	‡ **25.64**
	Zh	25.84	21.79	23.16	**23.56**	23.53
Ru	Ar	22.53	18.71	19.82	19.86	**20.35**
	Es	37.60	31.33	34.56	34.96	‡ **35.62**
	Fr	34.05	27.11	30.75	† 31.43	‡ **31.67**
	Zh	28.03	21.81	24.88	25.07	**25.12**
Zh	Ar	20.09	14.82	16.66	17.01	‡ **17.73**
	Es	30.66	23.15	27.84	27.99	**28.05**
	Fr	25.97	19.55	23.82	24.34	† **24.35**
	Ru	21.16	14.79	18.63	‡ 19.58	‡ **19.59**

Table 5.1: Results of each method, comparing the proposed syntactic matching methods in triangulation with other methods.

5.3.2 Results

Translation accuracy: The result of experiments using all combinations of pivot translation tasks for 5 languages via English is shown in Table 5.1. From the results, we can see that the proposed partial matching method of pivot subtrees in triangulation outperforms the standard triangulation method for all language pairs and achieves higher or almost equal scores than proposed exact matching method. The exact matching method also outperforms the standard triangulation method in the majority of the language pairs, but has a lesser improvement than partial matching method. Sequential pivot translation is uniformly weak than all the triangulation methods as shown in the previous research and Chapter 4.

Effect on coverage: In Table 5.2 we show the comparison of coverage of each

Source	Target	Number of source-side unique phrases/words	
		Tri. TreeExact	Tri. TreePartial
Ar	Es	2.580M / 5,072	2.646M / 5,077
	Fr	2.589M / 5,067	2.658M / 5,071
	Ru	2.347M / 5,085	2.406M / 5,088
	Zh	2.324M / 5,034	2.386M / 5,040
Es	Ar	1.942M / 5,182	2.013M / 5,188
	Fr	2.062M / 5,205	2.129M / 5,210
	Ru	1,978M / 5,191	2.037M / 5,197
	Zh	1,920M / 5,175	1.986M / 5,180
Fr	Ar	2.176M / 5,310	2.233M / 5,316
	Es	2.302M / 5,337	2.366M / 5,342
	Ru	2.203M / 5.311	2.266M / 5,318
	Zh	2.162M / 5.313	2.215M / 5,321
Ru	Ar	2.437M / 5,637	2.505M / 5,644
	Es	2.478M / 5.677	2.536M / 5,682
	Fr	2.479M / 5,661	2.531M / 5,665
	Zh	2.466M / 5,682	2.515M / 5,688
Zh	Ar	1.480M / 9,428	1.556M / 9,474
	Es	1.504M / 9,523	1.570M / 9,555
	Fr	1.499M / 9,490	1,568M / 9,520
	Ru	1.518M / 9,457	1.593M / 9,487

Table 5.2: Comparison of rule table coverage in proposed triangulation methods.

proposed triangulated method. From this table, we can see that the exact matching method reduces several percent in number of unique phrases while the partial matching method keeps the same coverage with surface-form matching. We can consider that it is one of the reasons of the difference in improvement stability between the partial and exact matching methods.

Noise reduction: The main motivation of using parse trees in the proposed methods is to prevent inappropriate connection of phrase correspondences and reduce the noises in rule table. To investigate how the syntactic matching methods success to remove noisy rules, we perform analysis of noise ratio. Noisy rules must contain source and target phrases having no correspondence in meaning, though we can not make a decision for all phrase pair candidates in rule tables. Therefore we assume that directly trained TMs with source-target parallel corpus

Source	Target	Noise Ratio in Triangulated Table [%]		
		Tri. Hiero	Tri. TreeExact	Tri. TreePartial
Ar	Es	78.40	**63.61** (-14.79)	68.51 (-9.88)
	Fr	81.39	**67.31** (-14.08)	72.22 (-9.17)
	Ru	81.87	**69.23** (-12.64)	73.84 (-8.03)
	Zh	75.70	**63.06** (-12.64)	67.72 (-7.98)
Es	Ar	80.03	**64.97** (-15.06)	69.80 (-10.23)
	Fr	81.55	**65.30** (-16.26)	70.61 (-10.94)
	Ru	81.45	**68.02** (-13.43)	72.67 (-8.78)
	Zh	74.05	**61.60** (-12.45)	65.90 (-8.15)
Fr	Ar	81.77	**67.69** (-14.08)	72.39 (-9.38)
	Es	80.94	**64.94** (-16.00)	70.20 (-10.74)
	Ru	82.77	**69.84** (-12.93)	74.52 (-8.25)
	Zh	76.14	**64.29** (-11.85)	68.53 (-7.61)
Ru	Ar	82.15	**70.15** (-12.00)	74.60 (-7.55)
	Es	79.80	**67.16** (-12.64)	71.68 (-8.12)
	Fr	82.41	**70.10** (-12.31)	74.76 (-7.65)
	Zh	76.07	**64.31** (-11.76)	68.67 (-7.40)
Zh	Ar	80.05	**66.90** (-13.15)	71.23 (-8.82)
	Es	77.94	**65.53** (-12.41)	69.70 (-8.24)
	Fr	78.24	**68.54** (-9.70)	72.51 (-5.73)
	Ru	79.80	**67.07** (-12.73)	71.27 (-8.53)

Table 5.3: Comparison of noise ratio in triangulated rule table

have fine approximation close to the ideal distribution of translation probability. To compute the noise ratio $noise(T_{tri}|T_{dir})$ of triangulated rule table T_{tri} with directly trained table T_{dir}, we define:

$$noise\left(T_{tri}|T_{dir}\right) = \frac{\sum_{\left(\overline{s},\overline{t}\right)\in T_{tri}\setminus T_{dir}} \phi\left(\overline{t}|\overline{s}\right)}{\sum_{\left(\overline{s},\overline{t}\right)\in T_{tri}} \phi\left(\overline{t}|\overline{s}\right)} \tag{5.9}$$

where $\phi(\overline{t}|\overline{s})$ is forward translation probability which can be considered as a most important feature of rule table. In Table 5.3, we show the calculated noise ratio of rule table for each triangulation method and language pair. This result shows that although triangulated rule tables contain many noisy rules, the syntactic matching methods indeed success to reduce them. Tri.TreeExact reduces up to -16.26% of noisy rules, and Tri.TreePartial reduces up to -10.94%. The reason

Source	Target	Distribution Error Rate (MAE / RMSE) [%]		
		Tri. Hiero	Tri. TreeExact	Tri. TreePartial
Ar	Es	14.16 / 10.62	14.49 / 11.06	13.96 / 10.56
	Fr	13.01 / 9.72	13.52 / 10.19	12.90 / 9.65
	Ru	12.64 / 9.51	12.33 / 9.24	12.03 / 8.97
	Zh	15.88 / 11.96	13.69 / 10.42	13.81 / 10.42
Es	Ar	13.90 / 10.29	13.84 / 10.30	13.44 / 9.92
	Fr	13.39 / 10.61	14.51 / 11.30	13.95 / 10.89
	Ru	12.81 / 9.71	12.92 / 9.70	12.52 / 9.38
	Zh	16.02 / 12.09	13.94 / 10.69	14.01 / 10.64
Fr	Ar	13.40 / 9.98	13.10 / 9.76	12.70 / 9.38
	Es	14.25 / 11.38	14.39 / 11.29	14.05 / 11.03
	Ru	12.58 / 9.58	12.46 / 9.37	11.98 / 8.99
	Zh	15.45 / 11.74	13.34 / 10.28	13.40 / 10.23
Ru	Ar	12.68 / 9.35	12.36 / 9.16	11.98 / 8.79
	Es	13.27 / 10.05	13.68 / 10.54	13.12 / 10.00
	Fr	12.29 / 9.28	12.84 / 9.78	12.13 / 9.17
	Zh	15.34 / 11.72	13.13 / 10.10	13.25 / 10.11
Zh	Ar	12.57 / 9.11	12.86 / 9.39	12.57 / 9.09
	Es	13.25 / 9.78	13.58 / 10.16	13.22 / 9.79
	Fr	12.86 / 9.49	12.67 / 9.44	12.25 / 9.07
	Ru	12.22 / 9.14	12.40 / 9.31	12.12 / 9.03

Table 5.4: Comparison of distribution error rate in triangulated rule table

why the noise reduction rate at Tri.TreePartial is smaller than Tri.TreePartial that Tri.TreePartial weakens the influence of noisy rules instead of removing them to keep the coverage.

Improvement of probability estimation: Although we see that syntactic matching methods help to reduce noisy rules, there is no guarantee that they can improve the estimation of translation probabilities. In Table 5.4, we show mean absolute error (MAE) and root mean square error (RMSE) for the distribution of forward translation probability scores of triangulated rule tables with directly trained rule tables. To calculate MAE and RMSE, we ignore the noisy rules which are not contained in directly trained rule tables, to separate different factors. From the result, we can see that Tri.TreePartial reduces MAE and RMSE, namely makes the distribution closer to the ideal, in almost language pairs. On the

other hand, Tri.TreeExact does not stably reduce them. It may be induced by the fact that the restricted matching condition of Tri.TreeExact excludes many unmatched phrase pair candidates and remove even translation rules which are not noisy. Therefore, we can consider that soften restriction of matching condition helps also to improve the estimation of translation probabilities.

Qualitative analysis: We show an example of a translated sentences for which pivot-side ambiguity is resolved in the syntactic matching methods:

Source Sentence in French:

La Suisse encourage **tous les États parties** à soutenir le travail conceptuel que fait actuellement le Secrétariat .

Corresponding Sentence in English:

Switzerland encourages all parties to support the current conceptual work of the secretariat.

Reference in Spanish:

Suiza alienta a **todos los Estados partes** a que apoyen la actual labor *conceptual* de la Secretaría .

Direct:

Suiza alienta a todos los Estados partes a que apoyen el trabajo conceptual que se examinan en la Secretaría . (BLEU+1: 55.99)

Tri. Hiero:

Suiza *conceptuales* para apoyar la labor que en estos momentos la Secretaría alienta a todos los Estados Partes . (BLEU+1: 29.74)

Tri. TreeExact:

Suiza alienta a **todos los Estados Partes** a apoyar la labor conceptual que actualmente la Secretaría . (BLEU+1: 43.08)

Tri. TreePartial:

Suiza alienta a **todos los Estados Partes** a apoyar la labor conceptual que actualmente la Secretaría . (BLEU+1: 43.08)

The results of Tri.TreeExact and Tri.TreePartial are same in this example. Digest of the derivation process in Tri. Hiero is:

$$
\begin{aligned}
S &\Rightarrow \langle X_0,\ X_0 \rangle \\
&\Rightarrow \langle \mathit{La\ Suisse}\ X_1,\ \mathit{Suiza}\ X_1 \rangle \\
&\Rightarrow \langle \mathit{La\ Suisse}\ X_2\ .,\ \mathit{Suiza}\ X_2\ . \rangle \\
&\Rightarrow \left\langle \mathit{La\ Suisse}\ \underline{X_3}\ \mathit{partie}\ \underset{\sim}{X_4}\ .,\ \mathit{Suiza}\ \underset{\sim}{X_4}\ \underline{X_3}\ \mathit{Partes}\ . \right\rangle \\
&\Rightarrow \ldots
\end{aligned}
$$

On the other hand, digest of the derivation process in Tri.TreeExact/Tri.TreePartial is:

$$
\begin{aligned}
S &\Rightarrow \langle X_0,\ X_0 \rangle \\
&\Rightarrow \langle \mathit{La\ Suisse}\ X_1,\ \mathit{Suiza}\ X_1 \rangle \\
&\Rightarrow \langle \mathit{La\ Suisse\ encourage}\ X_2\ X_3,\ \mathit{Suiza\ alienta\ a}\ X_2\ X_3 \rangle \\
&\Rightarrow \langle \mathit{La\ Suisse\ encourage\ tous\ les}\ X_4\ \mathit{partie}\ X_3,\ \mathit{Suiza\ alienta\ a\ todos}\ X_4\ \mathit{Partes}\ X_3 \rangle \\
&\Rightarrow \langle \mathit{La\ Suisse\ encourage\ tous\ les\ Etats\ partie}\ X_3,\ \mathit{Suiza\ alienta\ a\ todos\ los\ Estados\ Partes}\ X_3 \rangle \\
&\Rightarrow \ldots
\end{aligned}
$$

Here we can see that the derivation in Tri.Hiero uses rule $X \rightarrow \langle X_0$ parties X_1, $X_1\ X_0$ Partes$\rangle$[2] causing incorrect re-ordering of phrases followed by steps of incorrect word selection.[3] On the other hand, derivation in Tri.TreeExact and Tri.TreePartial uses rule $X \rightarrow \langle$tous les X_0 parties, todos X_0 Partes$\rangle$[4] synthesized from T2S rules with common pivot subtree (NP (DT all) (NP' X_{NNP} (NNS parties)). We can confirm that the derivation improves word-selection and word-reordering by using this rule.

[2]The words emphasized with underline and wavy-underline in the example correspond to X_0 and X_1 respectively.

[3]For example, the word "conceptuales" with italic face in Tri.Hiero takes the wrong form and position.

[4]The words emphasized in bold face in the example correspond to the rule.

vocabulary size:	16k (shared)
source embedding size:	512
target embedding size:	512
output embedding size:	512
encoder hidden size:	512
decoder hidden size:	512
LSTM layers:	1
attention type:	MLP
attention hidden size:	512
optimizer type:	Adam
loss integration type:	mean
batch size:	2048
max iteration:	200k
dropout rate:	0.3
decoder type:	Luong+ 2015

Table 5.5: Main parameters of NMT training

5.3.3 Comparison with Neural MT:

Recent results (Firat et al., 2016; Johnson et al., 2017) have found that neural machine translation systems can gain the ability to perform translation with zero parallel resources by training on multiple sets of bilingual data. However, previous work has not examined the competitiveness of these methods with pivot-based symbolic SMT frameworks such as PBMT or Hiero. In this section, we compare a zero-shot NMT model and other pivot NMT methods with our pivot-based Hiero models. To train and evaluate NMT models, we adopt NMTKit.[5] Detailed parameters to train NMT models are shown in Table 5.5.

We evaluate 4 additional translation methods:

Cascade NMT:

Sequential pivot translation with source-pivot and pivot-target NMTs (Section 3.1.1).

[5]https://github.com/odashi/nmtkit

Source	Target	BLEU Score [%]						
		Direct Hiero	Direct NMT	Tri. TreePartial	Cascade Heiro	Cascade NMT	Synthetic NMT	Zero-Shot NMT
Ar	Es	38.49	38.25	35.94	30.95	31.62	32.35	8.18
	Fr	33.34	33.16	30.83	25.08	26.91	29.51	8.57
	Ru	24.63	27.00	24.15	18.70	21.67	21.81	5.79
	Zh	27.27	30.04	25.07	21.77	23.70	25.63	5.04
Es	Ar	27.18	26.02	24.45	22.72	21.21	23.01	5.22
	Fr	43.24	41.83	40.12	35.40	31.84	36.57	15.04
	Ru	28.83	30.65	27.41	22.43	23.60	25.97	7.57
	Zh	27.08	32.36	25.16	23.36	26.03	27.31	8.62
Fr	Ar	25.10	23.28	22.13	19.88	18.66	18.83	8.08
	Es	45.20	44.49	41.99	37.75	32.93	36.78	14.37
	Ru	27.42	28.29	25.64	20.64	20.87	23.60	8.77
	Zh	25.84	29.10	23.53	21.79	23.14	24.96	11.95
Ru	Ar	22.53	23.19	20.35	18.71	19.71	19.21	3.18
	Es	37.60	38.67	35.62	31.33	31.25	31.22	10.42
	Fr	34.05	33.26	31.67	27.11	27.34	29.10	9.76
	Zh	28.03	31.39	25.12	21.81	24.25	25.46	9.46
Zh	Ar	20.09	20.17	17.73	14.82	16.89	18.01	10.38
	Es	30.66	32.69	28.05	23.15	26.01	27.80	6.13
	Fr	25.97	27.68	24.35	19.55	23.35	25.46	7.12
	Ru	21.16	23.17	19.59	14.79	18.40	20.53	3.21

Table 5.6: Comparison of SMT and NMT in multilingual translation tasks.

Synthetic NMT:

Generating pseudo-parallel corpus synthesized by translating pivot-side of source-pivot parallel corpus with pivot-target NMT (Section 3.1.2).

Zero-Shot NMT:

Training single shared model with pvt $\leftrightarrow$ {src,target} parallel data according to Johnson et al. (2017).

Direct NMT:

Translating with NMT directly trained on the source-target parallel corpus without using pivot language (for comparison).

Training data for Cascade NMT, Synthetic NMT, and Zero-Shot NMT are same

with Pivot Hiero methods (source-pivot and pivot-target corpora), and training data for Direct NMT is same with Direct Hiero.

In Table 5.6, we show BLEU score of each translation task and language pair. From the results we see the tendency of NMT that directly trained model achieves high translation accuracy even for translation between languages of different families, on the other hand, the accuracy is drastically reduced in the situation when there is no source-target parallel corpora for training. Among pivot and zero-shot methods for NMT, Synthetic NMT achieves the highest score for almost language pairs. The reason why Synthetic NMT outperforms Cascade NMT for the majority of language pairs, may be that multi-layer NNs have robustness for noisy training data, and can optimize the trained model with fine-tuning technique. On the other hand, for Cascade NMT, fine-tuning is available only for source-pivot and pivot-target TMs separately and not for the whole pipelined system.

In our setting, while bilingually trained NMT systems were competitive or outperformed Hiero-based models, zero-shot translation is uniformly weaker. This may be because we used only single LSTM layer for each of encoder and decoder, or because the amount of parallel corpora or language pairs were not sufficient. Thus, we can posit that while zero-shot translation has demonstrated reasonable results in some settings, successful zero-shot translation systems are far from trivial to build, and pivot-based symbolic MT systems such as PBMT or Hiero may still be a competitive alternative.

5.4 Summary

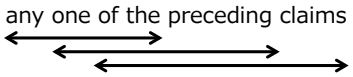
In this chapter, we have proposed a method of pivot translation using triangulation with exact or partial matching method of pivot-side parse subtrees. In experiments, we found that these triangulated models are effective in particular when allowing partial matching. From the analysis, we confirmed that the syntactic matching methods indeed help to reduce inappropriately connected rules and specifically partial matching method can stably improve the estimation of translation probabilities.

6 Syntactic and Non-Redundant Segment Selection for Active Learning

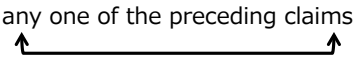
In Section 3.2, we showed the representative segment selection methods and their advantages and disadvantages, and mentioned problems of redundancy and fragmentation in segment selection phrase selection method using n-gram as shown in Figure 6.1 (a). In this chapter, we propose two methods that aim to solve these two problems and improve the efficiency and reliability of segment-based active learning for SMT. For the problem of overlapping phrases, we note that by merging overlapping phrases, as shown in Figure 6.1 (b), we can reduce the number of redundant words annotated and improve training efficiency. We adopt the idea of *maximal substrings* (Okanohara and Tsujii, 2009; Yamamoto and Church, 2001) which both encode this idea of redundancy. For the problem of phrase structure fragmentation, we propose a simple heuristic to count only *well-formed syntactic constituents* in a parse tree, as shown in Figure 6.1 (c).

6.1 Compact and Syntactically Coherent Segment Selection

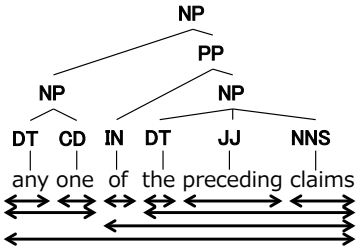
In this section, we explain the two proposed methods to solve the problems of redundancy and fragmentation and the combination method of them.



(a) Conventional n-gram selection method ($n = 4$)



(b) Proposed maximal phrase selection method



(c) Proposed parse subtree selection method

Figure 6.1: Conventional and proposed data selection methods

6.1.1 Segment Selection based on Phrase Maximality

To solve both the problem of overlapping phrases and the problem of requiring limits on phrase length for computational reasons, we propose a method using the idea of *maximal substrings* (Okanohara and Tsujii, 2009). Maximal substrings are formally defined as "a substring that is *not* always included in a particular longer substring." For example, if we define p_i as a phrase and $occ(p_i)$ as its occurrence count in a corpus, and have the following data:

$$\begin{aligned} p_1 &= \text{“one of the preceding”}, & occ(p_1) &= 200,000 \\ p_2 &= \text{“one of the preceding claims”}, & occ(p_2) &= 200,000 \\ p_3 &= \text{“any one of the preceding claims”}, & occ(p_3) &= 190,000 \end{aligned}$$

$p_1 =$ "one of the preceding" always co-occurs with the longer $p_2 =$ "one of the preceding claims" and thus is not a maximal substring. On the other hand, p_2 does not always co-occur with p_3, and thus p_2 will be maximal. This relationship

can be defined formally with the following semi-order relation:

$$s_1 \preceq s_2 \Leftrightarrow \exists \alpha, \beta : \ s_2 = \alpha s_1 \beta \wedge occ(s_1) = occ(s_2). \tag{6.1}$$

Demonstrating this by the previous example, $p_1 = \alpha p_2 \beta$, $\alpha =$ "", $\beta =$ "claims" hold, meaning p_1 is a sub-sequence of p_2, and p_2 is a sub-sequence of p_3 in a similar manner. Since p_1 is a sub-sequence of p_2 and $occ(p_1) = occ(p_2) = 200,000$, $p_1 \preceq p_2$ holds. However, although p_2 is a sub sequence of p_3, because $occ(p_2) = 200,000 \neq 190,000 = occ(p_3)$, the relation $p_2 \preceq p_3$ does not hold. Here, we say p has *maximality*[1] if there does not exist any q other than p itself that meets $p \preceq q$, and we call such a phrase a *maximal phrase*.

All the maximal phrases in the source language corpus with N words can be efficiently enumerated with linear time $O(N)$ by using enhanced suffix array (Kasai et al., 2001). Since it is required to compare strings $O(logN)$ times to obtain occurrence count of each maximal phrase for binary search, it is required to compare strings $O(NlogN)$ times for all the maximal phrases (Okanohara and Tsujii, 2009). Since the number of maximal phrases to be extracted is at most $N-1$, we can adopt sorting algorithm of computational complexity $O(NlogN)$ to enumerate them in descending order by frequency. In actual implementation of maximal phrase extraction, we exclude word strings containing newline characters, and extract only those with occurrence counts of 2 or more. This is to prevent a large number of word substrings including most sentences in the source language corpus from being selected as the maximal phrase of occurrence count 1.

To apply this concept to active learning, our proposed method limits translation data selection to only maximal phrases. This has two advantages. First, it reduces overlapping phrases to only the maximal phrases, allowing translators to cover multiple high-frequency phrases in the translation of a single segment. Second, it removes the need to set arbitrary limits on the length of strings such as $n = 4$ used in previous work.

However, it can be easily noticed that while in the previous example p_2 is included in p_3, their occurrence counts are close but not equivalent, and thus both are maximal phrases. In such a case, the naïve implementation of this method

[1]Maximality is a term of algebra, and $x \in S$ is called *maximal element* of a subset $S \subset P$ of some partially ordered set $(P, \preceq)$ if all $y \in S$, $x \preceq y$ implies $x = y$.

can not remove these redundant phrases, despite the fact that it is intuitively preferable that the selection method combines phrases if they have almost the same occurrence count. Thus, we also propose to use the following semi-order relation generalized with parameter λ:

$$s_1 \overset{*}{\preceq} s_2 \Leftrightarrow \exists \alpha, \beta : \ s_2 = \alpha s_1 \beta \wedge \lambda \cdot occ(s_1) < occ(s_2) \tag{6.2}$$

where λ takes a real numbered value from 0 to 1. We redefine maximality using this semi-order $\overset{*}{\preceq}$ as λ-*maximality*, and call maximal phrases defined with $\overset{*}{\preceq}$ λ-*maximal phrases* in contrast to standard maximal phrases. We propose a segment selection method in which we sequentially add highly frequent λ-maximal phrases uncovered in existing parallel corpus.

By setting the parameter λ of λ-maximal phrase selection to be smaller than 1, we can remove the restriction that the two segments under comparison be of exactly equal counts, allowing them to have only approximately the same occurrence count. As special cases, it is equivalent with standard maximal phrase selection when $\lambda = 1 - \epsilon$ (ϵ is a small positive number close to zero), and it becomes a random selection of full-sentences when $\lambda = 0$. In consideration of the possibility that both advantages can be compatible, we set to $\lambda = 0.5$, which is an intermediate value between two special values, for comparison with other methods in this research. By using λ-maximal phrases with $\lambda = 0.5$ instead of standard maximal phrases, we can remove a large number of phrases that are included in a particular longer phrase more than half the time, indicating that it might be preferable to translate the longer phrase.

Since λ-maximal phrases with $\lambda < 1$ always satisfies the condition of standard maximal phrase, all candidates for λ-maximal phrase in the source language corpus can be searched among all standard maximal phrases.

6.1.2 Phrase Selection based on Parse Trees

In this section, we propose a second phrase selection method based on the results from the syntactic analysis of source language data. This method first processes all the source language data with a phrase structure parser, traverses and counts up all the subtrees of parse trees as shown in Figure 6.2, and finally selects

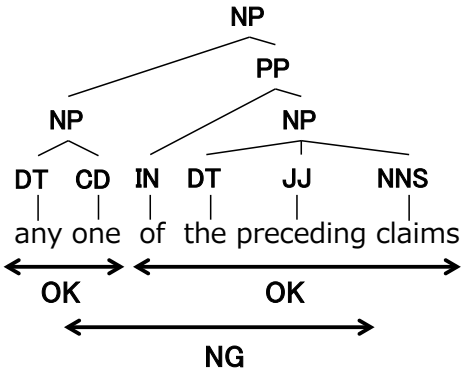


Figure 6.2: Phrase selection based on parse trees

phrases corresponding to a subtree in frequency order. We propose this method because we expect the selected phrases to have syntactically coherent meaning, potentially making human translation easier than other methods that do not use syntactic information. In this research, we aim to investigate the influence of selected phrases on active learning efficiency, and thus we use parse trees only for phrase extraction so that it is easier to compare with other phrase selection methods.

Since the phrase selection methods based on n-gram or phrase maximality count phrases as superficial sequences of words, for example if we have a phrase "are proposed and discussed", the methods count up also the sub sequence of words "are proposed", while the proposed method based on parse trees does not select that because of fragmentation. Therefore, the occurrence counts by this syntactic method tend to be lower than superficial phrase counting, and in consequence the priority that phrases with length 2 or more words are selected tends to be lower than single words.

It should be noted that because this method counts all subtrees, it is capable of selecting overlapping phrases like the methods based on n-grams. Therefore we also experiment with a method using together both subtrees and the λ-maximal phrases proposed in Section 6.1.1 to select both syntactic and non-redundant segments.

6.2 Simulation Experiment

6.2.1 Experimental Set-Up

To investigate the effects of the phrase selection methods proposed in Section 6.1, we first performed a simulation experiment in which we incrementally retrain translation models and evaluate the accuracy after each step of data selection. In this experiment, we chose English as a source language and French and Japanese as target languages. To simulate a realistic active learning scenario, we started from given parallel data in the general domain and sequentially added additional source language data in a specific target domain. For the English-French translation task, we adopted the Europarl corpus[2] (Koehn, 2005) from WMT2014[3] as a base parallel data source and EMEA[4] (Tiedemann, 2009), PatTR[5] (Wäschle and Riezler, 2012), and Wikipedia titles, used in the medical translation task, as the target domain data. For the English-Japanese translation task, we adopted the broad-coverage example sentence corpus provided with the Eijiro dictionary[6] as general domain data, and the ASPEC[7] (Nakazawa et al., 2016) scientific paper abstract corpus as the target domain data. For pre-processing, we tokenized Japanese corpora using the KyTea word segmenter (Neubig et al., 2011) and filtered out the lines of length over 60 from all the training parallel data to ensure accuracy of parsing and alignment. We show the details of the parallel dataset after pre-processing in Table 6.1.

For the machine translation framework, we used phrase-based SMT (Koehn et al., 2003) with the Moses toolkit[8] (Koehn et al., 2007) as a decoder. To efficiently re-train the models with new data, we adopted inc-giza-pp,[9] a specialized version of GIZA++ word aligner (Och and Ney, 2003) supporting incremental training, and the memory-mapped dynamic suffix array phrase tables (MMSAPT) feature of Moses (Germann, 2014) for on-memory construction of phrase tables.

[2]http://www.statmt.org/europarl/
[3]http://statmt.org/wmt14/
[4]http://opus.lingfil.uu.se/EMEA.php
[5]http://www.cl.uni-heidelberg.de/statnlpgroup/pattr/
[6]http://eijiro.jp
[7]http://lotus.kuee.kyoto-u.ac.jp/ASPEC/
[8]http://www.statmt.org/moses
[9]https://github.com/akivajp/inc-giza-pp

Lang Pair	Domain	Dataset	Amount
En-Fr	General (Base)	Train	1.89M Sent. En: 47.6M Words Fr: 49.4M Words
	Medical (Target)	Train	15.5M Sent. En: 393M Words Fr: 418M Words
		Test	1000 Sent.
		Dev	500 Sent.
En-Ja	General (Base)	Train	414k Sent. En: 6.72M Words Ja: 9.69M Words
	Scientific (Target)	Train	1.87M Sent. En: 46.4M Words Ja: 57.6M Words
		Test	1790 Sent.
		Dev	1790 Sent.

Table 6.1: Details of parallel data

We train 5-gram models over the target side of all the general domain and target domain data using KenLM (Heafield, 2011). For the tuning of decoding parameters, since it is not realistic to run MERT (Och, 2003) at each retraining step, we tuned the parameters to maximize the BLEU-4 score (Papineni et al., 2002) for the baseline system, and re-used the parameters thereafter. We compare the following 8 segment selection methods, including 2 random selection methods, 2 conventional methods and 4 proposed methods:

sent-rand:

Select sentences randomly.

4gram-rand:

Select n-gram strings with length of up to 4 in random order.

sent-by-4gram-freq:

Select the sentence including the most frequent uncovered phrase with length of up to 4 words (baseline 1, Section 3.2.2).

4gram-freq:

Select the most frequent uncovered phrase with length of up to 4 words (baseline 2, Section 3.2.3).

maxsubst-freq:

Select the most frequent uncovered maximal phrase (proposed, Section 6.1.1)

reduced-maxsubst-freq:

Select the most frequent uncovered λ-phrase with $\lambda = 0.5$ (proposed, Section 6.1.1)

struct-freq:

Select the most frequent uncovered phrase extracted from the subtrees (proposed, Section 6.1.2).

reduced-struct-freq:

Select the most frequent uncovered λ-maximal phrase ($\lambda = 0.5$) extracted from the subtrees (proposed, Section 6.1.1 and Section 6.1.2).

To generate oracle translations, we used an SMT system trained on all of the data in both the general and target-domain corpora. To generate parse trees, we used the Ckylark parser (Oda et al., 2015).

6.2.2 Results and Discussion

Comparison of efficiency: From the results obtained by simulation experiments, Table 6.2 shows the transition of BLEU scores for each method at the time of adding no words, 10k words, 100k words, 1M words, and all segments.[10] The score at the time of adding all segments is considered to be the performance

[10] BLEU score of the base system in English-Japanese translation started from a low value lower than 10 because the domain-specific parallel sentences are extremely short. In the previous research (Ananthakrishnan et al., 2010a,b; Bloodgood and Callison-Burch, 2010; Haffari et al., 2009), BLEU score is commonly used in domain adaptation with active learning from the situation where the parallel sentences are insufficient. Therefore, we used the same evaluation measure in this research.

Lang Pair	Selection Method	BLEU-4 Score [%]				
		No Addition	10k Words	100k Words	1M Words	All Segments
En-Fr	sent-rand	25.39	25.57	25.72	27.35	30.02
	4gram-rand		25.53	25.52	27.16	28.32
	sent-by-4gram-freq		25.55	26.12	<u>27.93</u>	30.69
	4gram-freq		<u>25.61</u>	<u>26.16</u>	27.89	28.75
	maxsubst-freq		25.55	25.84	27.49	29.60
	reduced-maxsubst-freq		**25.63**	26.10	27.91	29.81
	struct-freq		**25.85**	**26.86**	**29.06**	30.03
	reduced-struct-freq		† **26.08**	† **27.18**	† **29.40**	30.20
En-Ja	sent-rand	9.37	10.44	13.03	15.58	21.22
	4gram-rand		10.57	13.37	17.61	19.71
	sent-by-4gram-freq		11.14	14.49	17.66	21.06
	4gram-freq		<u>11.49</u>	<u>15.07</u>	<u>18.27</u>	19.74
	maxsubst-freq		**11.72**	**15.13**	**18.58**	19.88
	reduced-maxsubst-freq		**11.87**	† **15.72**	**18.71**	19.59
	struct-freq		**12.02**	**15.44**	**18.61**	19.97
	reduced-struct-freq		† **12.27**	**15.66**	† **18.91**	19.83

Table 6.2: Transition of BLEU score according to the number of additional words. Underlines indicate that the score is the maximum among the random selection method and the baseline method at each point in time immediately after the addition of 1M words. Bold face indicates the scores the proposed method exceeding the underlined scores. Daggers † indicate the best score in all the methods in each stage.

limit of each method, since this is the translation accuracy by using all the translated segments. However, since the number of source words to be added varies significantly, comparison can not be made simply from the viewpoint of active learning efficiency.

Comparing two random selection methods and the two baseline methods form the table, the accuracy elongation with 4gram-freq is stably high until addition of 100k words. Accordingly, we can confirm the advantages of selecting highly frequent segments instead of whole sentences. However, at the time of addition of 1M words in English-French translation, the score of 4gram-freq is lower than sent-by-4gram-freq, and the score of 4gram-freq is lower than sent-by-4gram-freq and sent-rand at the time of addition of all segments in both language pairs. From this fact, it can be seen that in the case of adding a number of words

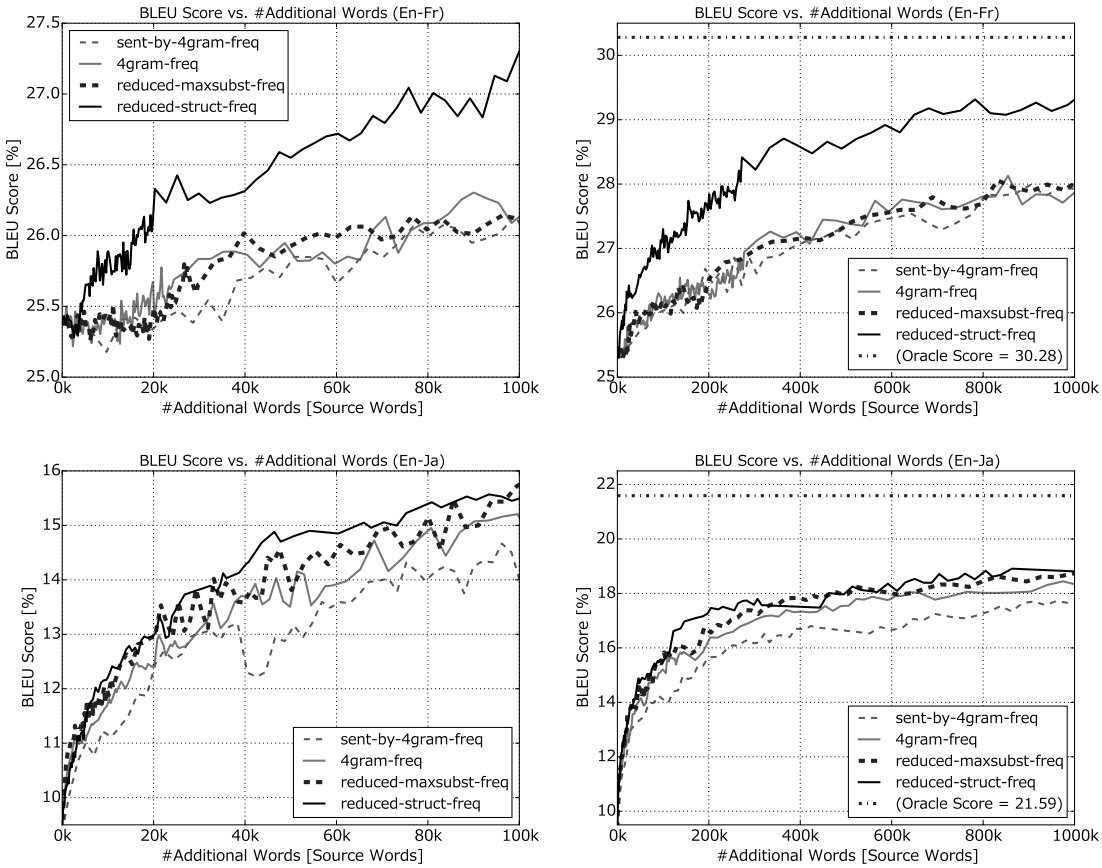


Figure 6.3: BLEU score vs. number of additional source words in each method (upper: En-Fr translation task, bottom: En-Ja translation task, left: up to 100k additional words, right: up to 1M additional words)

more than a certain amount, active learning efficiency of 4gram-freq is lower than sentence selection methods, and its performance limit is not high. It may because the length of selected segments is limited to 4 words in 4gram-freq as mentioned in Section 3.2.3, and the inability to learn correspondence of longer segments is disadvantage in MT.

Next, we compare the proposed methods with the baseline methods. In the proposed methods, reduced-maxsubst-freq is almost always higher than maxsubst-freq, and reduced-struct-freq is almost always higher than struct-freq. Thereby, it is considered that coverage is improved with fewer additional words by giving

priority to longer segments based on λ-maximality. Therefore, we compare the two proposed methods based on λ-maximality with two baseline methods in more detail.

In Figure 6.3, we show the evaluation score results by the number of additional source words up to 100k and 1M words. Additionally, we show the score of MT trained and evaluated using the whole data of the base corpus and the additional corpus as an oracle score, together with the right-side graph of up to the 1M additional words.

In reduced-maxsubst-freq, English-Japanese translation score was stably higher than the baseline methods. However in English-French translation, it got almost the same score as 4gram-freq up to the point of adding 1M words. However, since the score at the time of adding all the segments in both language pairs greatly exceeds 4gram-freq and 4gram-rand, the performance limit of this method is high. This seems to be a major reason that the problem of the maximum phrase length limitation as described above does not occur in the proposed methods. To investigate the reason why there is no significant difference between reduced-maxsubst-freq and 4gram-freq in English-French translation, we look over the highly frequent segments selected by both methods. The most frequent uncovered segments "according to claim" (1,502,455 times), "claim 1" (1,133,243 times), "characterized in that" (858,404 times), etc. were commonly selected in the two methods, and redundant segments as described in Section 3.2.3 were few, but rather fragmentation of phrase structures was conspicuous. A state in which multiple frequent 4-gram segments have many common word sub-sequence occurs particularly when frequent segments including more than 4 words. It seems to be because such frequent longer segments in the medical-domain corpus used in this experiment does not contain much domain-specific expressions, and they are already included in the large-scale general-domain corpus. On the other hand, in English-Japanese translation, many duplicates are seen in such as highly frequent segments "results suggest that" (6,352 times), "these results suggest" (5,115 times), "these results suggest that" (4,791 times), etc. selected by 4gram-freq. In this situation, reduced-maxsubst-freq can demonstrate higher learning efficiency by combining such overlapping segments into one. In particular, since the general-domain corpus used in English-Japanese translation is comparatively

Lang Pair	Selection Method	All Selected Segments			First 10k Words	
		#Segments	#Words	Average Phrase Length	#Segments	Average Segment Length
En-Fr	sent-by-4gram-freq	10.6M	269M	25.4	310	32.1
	4gram-freq	40.1M	134M	3.34	3.62k	2.76
	maxsubst-freq	62.4M	331M	5.30	2.39k	4.17
	reduced-maxsubst-freq	45.9M	246M	5.36	2.95k	3.39
	struct-freq	14.1M	94.2M	6.68	4.01k	2.49
	reduced-struct-freq	7.33M	41.3M	5.63	4.55k	2.20
En-Ja	sent-by-4gram-freq	1.28M	33.6M	26.3	560	17.8
	4gram-freq	8.48M	26.0M	3.07	4.70k	2.13
	maxsubst-freq	7.29M	25.8M	3.54	4.51k	2.22
	reduced-maxsubst-freq	6.06M	21.7M	3.58	4.76k	2.10
	struct-freq	1.45M	4.85M	3.34	6.64k	1.51
	reduced-struct-freq	1.10M	3.33M	3.03	6.73k	1.49

Table 6.3: Number of segments and average words/segment in each method

small scale of 400k sentences which summarizes daily expressions, frequent long segments in the scientific-domain corpus are not much covered.

In both language pairs, reduced-struct-freq almost stably achieves the highest score among all the segment selection methods. Only at the time of adding 100k words in English-Japanese translation, the score of reduced-maxsubst-freq is the maximum, though it is a small difference with reduced-maxsubst-freq, and the fluctuation range of the learning curve is large, so it seems to be within the error range[11]. We can see that in English-French translation, the scores of reduced-struct-freq and struct-freq based on parse trees grows more rapidly than other methods and is significantly better even at the point of 1M additional words. Besides in English-Japanese translation, the scores of these methods do not have much difference with reduced-maxsubst-freq and 4gram-freq while the number of additional words is small, but become higher than the other methods from the point of adding about 40k words. From the point of adding 500k words in English-Japanese translation, the score of each selection method becomes almost flat.

[11] When we tested the statistically significant difference by bootstrap resampling method (Koehn, 2004), no significant difference of $p < 0.1$ was observed.

Length of selected phrases: Due to the different criteria used by each method, there are also significant differences in the features of the selected phrases. In Table 6.3, we show the details of the number of all selected phrases, words and average phrase length until the stop condition, and at the point of 10k additional source words. The coverage converges when all the selected segments are translated, and thus the less words the method selects finally, the more rapidly the coverage converges and the translation accuracy may also improves. Whereas, since longer phrase can cover more n-gram phrases at once, methods selecting longer segments on the average have an advantage for improving 4-gram coverage. We see there is a big difference of average phrase length between 5.30-6.68 in English-French and 3.03-3.58 in English-Japanese, but this depends only on the combination of source-side base and additional datasets and of course does not depend on language pairs. Moreover, we can confirm that the average phrase length at the point of 10k additional words in Table 6.3 is shorter than adding all the candidates especially in methods based on parse trees. That is clear because shorter phrases tend to be more frequent and selected earlier, still we see the tendency that the frequency of longer syntactic phrases drastically decreases.

Here we see the tendency that the selection methods based on parse trees select shorter phrases than other methods. This is caused by the fact that longer phrases are only counted if they cover a syntactically defined phrases, and thus longer substrings that do not form syntactic phrases are removed from consideration.

Effect on coverage: This difference in the features of the selected phrases also affects how well they can cover new incoming test data. To demonstrate this, in Table 6.4 we show the 1-gram and 4-gram coverage of the test dataset after 10k, 100k and 1M words have been selected. From the results, we can see that the reduced-struct-freq method attains the highest 1-gram coverage, efficiently covering unknown words. On the other hand, it is clear that methods selecting longer phrases have an advantage for 4-gram coverage, and we see the highest 4-gram coverage in the sent-by-4gram-freq method. In English-French translation, there is no change in the top four digits of 4-gram coverage at the time of adding 10k words. Although, whether to select a long segment or a short segment causes a trade-off relationship considering the effect on coverage, it was confirmed that 1-gram coverage and 4-gram coverage can be jointly improved by eliminating

Lang Pair	Selection Method	1-gram / 4-gram Coverage [%]			
		No Addition	10k Words	100k Words	1M Words
En-Fr	sent-rand	92.72 / 10.60	92.93 / 10.60	93.73 / 10.71	95.94 / 11.30
	4gram-rand		92.95 / 10.60	93.99 / 10.60	96.42 / 10.64
	sent-by-4gram-freq		92.95 / 10.60	93.96 / **10.72**	96.25 / **11.55**
	4gram-freq		92.92 / 10.60	94.46 / 10.66	96.60 / 11.16
	maxsubst-freq		92.79 / 10.60	93.61 / 10.62	95.99 / 10.92
	reduced-maxsubst-freq		92.92 / 10.60	94.38 / 10.66	96.55 / 11.13
	struct-freq		93.63 / 10.60	96.15 / 10.65	97.84 / 11.28
	reduced-struct-freq		**94.02** / 10.60	**96.38** / 10.69	**98.00** / 11.38
En-Ja	sent-rand	94.36 / 5.38	94.81 / 5.63	95.99 / 6.59	97.54 / 10.06
	4gram-rand		94.80 / 5.38	96.10 / 5.46	97.67 / 5.98
	sent-by-4gram-freq		95.10 / 5.84	96.28 / **7.23**	97.64 / **11.39**
	4gram-freq		95.64 / 5.97	96.87 / 7.14	97.97 / 10.43
	maxsubst-freq		95.59 / 5.96	96.83 / 7.07	97.91 / 10.20
	reduced-maxsubst-freq		95.73 / **6.00**	96.97 / 7.19	98.00/10.57
	struct-freq		96.60 / 5.44	97.80 / 5.79	98.58 / 7.02
	reduced-struct-freq		**96.64** / 5.44	**97.84** / 5.80	**98.61** / 7.14

Table 6.4: Effect on coverage in each selection method (rounded off to the second decimal place). Bold face indicates the highest coverage for each number of additional words.

duplication based on λ-maximality.

Reduction effect: In Section 3.2.3, as a matter of 4gram-freq, we mentioned that there are many common word sub-sequences appearing duplicated among selected segments, and to deal with this problem, we proposed selection method based on λ-maximality in Section 6.1.1. Table 6.5 summarizes the number and percentage of words which are selected by 4gram-freq and reduced by combining into longer segments with maxsubst-freq and reduced-struct-freq at the time of adding 10k words, 100k words and 1M words. From the table, it can be seen that in both language pairs, maxsubst-freq can reduce only a small amount of 1%-4%. As stated in Section 6.1.1, it is because standard phrase maximality has severe restrictions that the occurrence counts of segments in inclusion relation are equivalent, and thus many included word sub-sequences become maximal phrases. On the other hand, in both language pairs, reduced-maxsubst-freq reduce words by at least 24.70% up to 50.79%. From this results, it can be said that it is possible to effectively reduce the number of included phrases by relaxing the

Lang Pair	Selection Method	Number of Reduced Words (Reduction Ratio)		
		10k Add. Words	100k Add. Words	1M Add. Words
En-Fr	maxsubst-freq	92 (0.92%)	2,077 (2.11%)	34,917 (3.49%)
	reduced-maxsubst-freq	5,079 (50.79%)	42,622 (42.62%)	378,938 (37.89%)
En-Ja	maxsubst-freq	138 (1.38%)	686 (1.61%)	41,046 (4.10%)
	reduced-maxsubst-freq	2,560 (25.6%)	24,697 (24.70%)	24,697 (24.70%)

Table 6.5: Reduction amount of duplicated segments selected in 4gram-freq

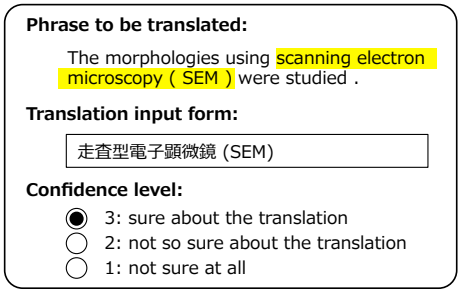


Figure 6.4: Example of the human translation interface

matching condition of occurrence counts of word sequences.

6.3 Manual Translation Experiment

6.3.1 Experimental Set-Up

To confirm that the results from the simulation in the previous section carry over to actual translators, we further performed experiments in which professional translators translated the selected segments. This also allowed us to examine the actual amount of time required to perform translation, and how confident the translators were in their translations.

We designed a web user interface as shown in Figure 6.4, and outsourced to an external organization that had three professional translators translate the shown phrases. As is standard when hiring translators, we paid a fixed price per word translated for all of the methods. Because showing only the candidate phrase out of context could cause difficulty in translation, we followed

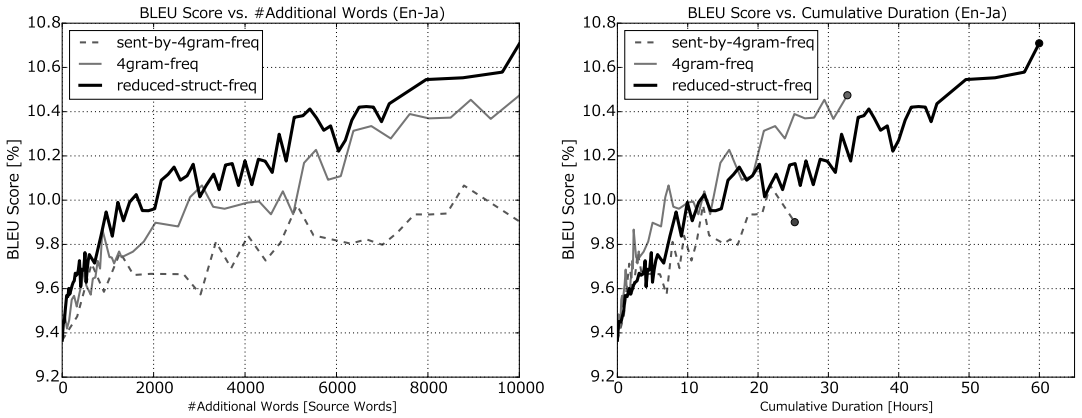


Figure 6.5: Transition of BLEU score vs. additional source words (left) and vs. cumulative working duration (right)

Bloodgood and Callison-Burch (2010) in showing a sentence including the selected phrase,[12] highlighting the phrase, and requesting to translate the highlighted part. We also requested every worker to select from 3 levels indicating how confident they were of their translation. In the background, the time required to complete the translation was measured from when the new phrase was shown until when the translation was submitted.

The methods selected for comparative evaluation are sentence selection based on 4-gram frequency (sent-by-4gram-freq) and phrase selection based on 4-gram frequency (4gram-freq) as baseline methods, and the phrase selection based on both parse trees and λ-maximality (reduced-struct-freq) as the proposed method. For each method we collected translations of 10k source words, alternating between segments selected by each method to prevent bias.

We used the same dataset as the English-Japanese translation task and the same tools in the simulation experiment (Section 6.2). However, for training target language models, we interpolated one trained with the base data and a second trained with collected data by using SRILM (Stolcke, 2002) because the hand-made data set was too small to train a full language model using only this data[13]. We tuned the interpolation coefficient such that it maximizes the

[12]Specifically, we selected the shortest sentence including the phrase in the source corpus.

[13] Let $P_L(e)$ be the occurrence probability of sentence e in language model L, then the occurrence

Selection Method	Total Working Time [Hours]	Average Confidence Level (3 Levels)	Percentage of Confidence Level			
			Skipped	Level 1	Level 2	Level 3
sent-by-4gram-freq	25.22	2.689	1.77%	0.00%	30.74%	69.08%
4gram-freq	32.70	2.601	0.53%	1.69%	35.48%	62.29%
reduced-struct-freq	59.97	**2.771**	0.52%	1.51%	18.82%	**79.15%**

Table 6.6: Total working time and statistics of confidence level evaluation

perplexity for the tuning dataset.

6.3.2 Results and Discussion

Efficiency results: Figure 6.5 shows the evaluation scores of SMT systems trained using varying amounts of collected phrases. In the left graph, we see the proposed method based on parse trees and phrase λ-maximality rapidly improves BLEU score, and requires fewer additional words than the conventional methods. Because the cost paid for translation often is decided by the number of words, this indicates that the proposed method has better cost performance in these situations. The right graph shows improvement by the amount of translation time. These results here are different, showing the 4-gram-freq baseline slightly superior. As discussed in Table 6.4, the methods based on parse trees select more uncovered 1-grams, namely unknown words, and specifically the proposed method selected more technical terms that took a longer time to translate.

Working time and confidence: We show the total time to collect the translations of 10k source words and statistics of confidence level for each method in Table 6.6. The total working time for the proposed method is nearly double that of other methods, and the tendency to focus on selecting uncovered technical terms as described above can been confirmed. This tendency is seen also by that the number of selected single words is nearly four times that of 4gram-freq,

probability in interpolated language model L_{1+2} from L_1 and L_2 is $P_{L_{1+2}}(e) = \alpha P_{L_1}(e) + (1-\alpha)P_{L_2}(e)$. Interpolation coefficient α takes a range from 0 to 1, and is adjusted with respect to the development data E_{dev} to minimize $PPL(E_{dev}) = exp\left(\frac{\sum_{e \in E_{dev}} -logP_{L_{1+2}}(e)}{|E_{dev}|}\right)$.

	Number of Segments					
Selection Method	1 Word	2 Words	3 Words	4 Words	5+ Words	Total
sent-by-4gram-freq	-	-	-	-	565	566
4gram-freq	1,185	2,061	1,045	390	0	4,681
reduced-struct-freq	4,688	1,038	884	96	38	6,744

Table 6.7: Detail of selected segments by length

	Average Working Time [Seconds]				
Selection Method	1 Word	2 Words	3 Words	4 Words	5+ Words
sent-by-4gram-freq	-	-	-	-	160.64
4gram-freq	30.14	24.76	21.77	21.12	-
reduced-struct-freq	35.61	25.23	21.72	28.13	22.82

Table 6.8: Average working time of manual translation corresponding to segment length

and the total number of selected segments is also large, as shown in Table 6.7. On the other hand, the segments selected by the proposed method were given the highest confidence level, receiving the maximum value of 3 for about 79% of phrase pairs, indicating that the generated parallel data is of high quality. To some extent, this corroborates our hypothesis that the more syntactic phrases selected by the proposed method are easier to translate.

We can also examine the tendency of working time for segments of different lengths in Table 6.8. Interestingly, single words consistently have a longer average translation time than phrases of length 2-4, likely because they tend to be technical terms that require looking up in a dictionary. In addition, since these are the average time required for translation of single segment, substantial average time cost for single word translation is more than double the word translation time of segments with 2 words, we can see how expensive the cost of translating technical terms.

We show the average confidence levels corresponding to phrase length in Table 6.9. The confidence level of single words in the proposed method is lower than in the baseline method, likely because the baseline selects a smaller amount of

Selection Method	Average Confidence Level (3 Levels)				
	1 Word	2 Words	3 Words	4 Words	5+ Words
sent-by-4gram-freq	-	-	-	-	2.689
4gram-freq	2.885	2.585	2.422	2.300	-
reduced-struct-freq	2.802	2.796	2.778	2.708	2.737

Table 6.9: Average confidence level of manual translation corresponding to phrase length

Selection Methods	BLEU Score [%]		
	Confidence 1+ (All)	Confidence 2+	Confidence 3
sent-by-4gram-freq	9.88	9.92	9.85
4gram-freq	10.48	10.54	10.36
reduced-struct-freq	10.70	**10.72**	10.67

Table 6.10: BLEU score when training on phrases with a certain confidence level

single words, and those selected are less likely to be technical terms. On the other hand, we can confirm that the confidence level for longer phrases in the baseline method decreases drastically, while it is stably high in our method, confirming the effectiveness of selecting syntactically coherent phrases.

Translation accuracy by confidence level: Finally, we show the accuracy of the SMT system trained by all the collected data in each method in Table 6.10. To utilize the confidence level annotation, we tested SMT systems trained by phrase pairs with confidence levels higher than 2 or 3. From the results, the accuracy of every method is improved when phrases pairs with confidence level 1 were filtered out. In contrast, the accuracy is conversely degraded if we use only phrase pairs with confidence level 3. The translation accuracy of 9.37% BLEU with the base SMT system without additional data became 10.72% after adding phrase pairs having confidence level 2 or higher, allowing for a relatively large gain of 1.35 BLEU points.

6.4 Summary

In this chapter, we proposed a new method for active learning in machine translation that selects syntactic, non-redundant phrases using parse trees and λ-maximal phrases. We first performed simulation experiments and obtained improvements in translation accuracy with fewer additional words. Further manual translation experiments also demonstrated that our method allows for greater improvements in accuracy and translator confidence.

7 Conclusion

7.1 Contribution

In SMT and NMT, it has been observed that translation with models trained on larger parallel corpora can achieve higher accuracy, and usually millions of sentence pairs are required in order to produce a high quality translation system. Unfortunately, readily available parallel corpora are limited for most language pairs, particularly those that do not include English, having few sentence pairs, or none at all.

In this thesis, we focused on the low-resource data scenario for MT, in which the size of the bilingual corpus is known to be limited. Specifically, our methods addressed to improve MT quality with two types of common approaches for coping with the scarceness of bilingual corpus: (1) pivot translation and (2) active learning for MT. Indeed, as we have demonstrated, MT quality could significantly benefit from syntactic and contextual information when faced with limited training data.

Contextual Disambiguation with Pivot-Side Language Models:

In Chapter 4, we proposed a new method in pivot translation to resolve pivot-side contextual ambiguity. This proposed method lets MT models remember the information of the pivot phrase. This information can help to select appropriate translation rules considering pivot-side context with pivot language models. Experimental results on multilingual translation showed significant improvement of MT evaluation scores for all the tested language pairs with relatively small indirectly parallel corpora, of 100k sentence pairs, and large English monolingual corpus as an additional resource. This method is effective in the case that available source-pivot and pivot-target parallel corpora are not large and while the amount of available pivot monolingual corpus is large.

Syntactic Disambiguation with Pivot-Side Parse Trees:

In Chapter 5, we proposed a new method in pivot translation to resolve the pivot-side syntactic ambiguity. This proposed method introduces an explicitly syntax-aware matching condition to find correct correspondences between source-pivot and pivot-target translation rules, and can produce more reliable models. Experimental results on multilingual translation showed significant improvements of MT evaluation scores for all the tested language pairs with larger indirectly parallel corpora than Chapter 4, of 1M sentence pairs, and results of English syntactic parsing as an additional resource. A syntactic matching method allowing partial matching successfully reduced the number of noisy translation rules and improved the estimation of translation probabilities causing better translation accuracy. This method is effective in the case that accurate syntactic parsers for the pivot language are available, and practical to use for pivot translation with larger amounts of parallel corpora.

Cost Reduction and Quality Improvement in Human Translation:

In Chapter 6, we proposed a new method in active learning for SMT to introduce new criteria for segment selection, based on non-redundancy and syntactic coherence. This proposed method provides a more compact and human-friendly annotation task than conventional methods, resulting in a higher quality parallel corpus with lower annotation cost. Experimental results on multilingual translation showed a significant improvement in all the tested language pairs. Experiments using both simulation and extensive manual translation by professional translators find the proposed method effective, achieving both greater gain of translation score for the same number of translated words, and allowing translators to be more confident in their translations.

7.2 Future Directions

Here, we list directions for future work in low-resource MT.

Pivot Translation Preserving Linguistic Information:

The proposed method in pivot translation in Chapter 4 uses MSCFG models that have potential to various information of source, target, and pivot languages.

For example, we should be able to combine the proposed methods in Chapters 4-5 and let to MSCFG model to remember the pivot tree structures.

As a more advanced method, it should be possible to devise compounded MSCFG models that can store not only pivot-side syntactic information but also source-side syntactic information, thereby realizing translation with higher reproducibility of source information. We mentioned that pivot translation has the problem of losing source language information, affected by the expressiveness of pivot language. In fact, this problem often occurs not only for MT, but also for human translators. For example, since modern English is known for its simple morphology which has no complicated grammatical conjugation such as personal suffixes, linguistic modality such as number, case, gender, etc. This information is lost when translating into English, resulting that translation from English into another language is different from the original meaning. In this method, we aim to achieve translation that preserves originally linguistic information by combining with pivot-side syntactic structures.

Active Learning for Pivot MT:

In many cases, pivot translation approach may solve the scarceness problem of bilingual corpora for many language pairs that have sufficient amount of indirectly parallel corpus with English. However, pivot translation alone has poor performance or no effect at all for truly low-resourced language pairs that have no suitable candidate to be a pivot with a significantly large parallel corpus. In such a case, an active learning approach should have a synergy with pivot translation, and be able to be used to efficiently supplement a shortage in data that can be used in pivot translation. Therefore, it is important to combine the approaches of pivot translation and active learning to provide a realistic solution in low-resource scenarios. It is natural that source-pivot, pivot-target and source-target language pairs have different annotation costs thereby optimization will become more complicated. Tackling this is an interesting problem.

Toward Multilinguality and Multimodality:

In recent years, research on NMT to utilize multiple available training data such as multi-source NMT (Zoph and Knight, 2016) and multimodal NMT (Specia et al., 2016) has become active and diversified. The previous work has demon-

strated the capability of multi-layer NNs to jointly train shared models. One final challenge for the future is to design an ecosystem of NNs by combining pivot MT and active learning with multi-source NMT. As mentioned previously, even today, many language pairs do not have a sufficient amount of parallel data, though this thesis demonstrates that we can efficiently grow the parallel corpus by applying active learning methods. Of course, the obtained parallel corpus can be used for training a single MT model or applying pivot translation in a straightforward manner. However, NMT models should be capable of reusing human annotation results as an additional source, and reuse even their own output. In this idea, the model will be able to train itself by feeding not only a regular parallel corpus, also from human translation results, output of other MT systems or of itself, and other possible resources such as different kind of human annotation or other language data. By applying this idea, the model can assist a human translator in such a manner of post-editing, and immediately adapt itself with the result, and even will be capable of automatic post-editing, resulting a substantial reduction of translation cost.

Bibliography

Sankaranarayanan Ananthakrishnan, Rohit Prasad, David Stallard, and Prem Natarajan. A Semi-Supervised Batch-Mode Active Learning Strategy for Improved Statistical Machine Translation. In *Proc. CoNLL*, pages 126–134, 2010a.

Sankaranarayanan Ananthakrishnan, Rohit Prasad, David Stallard, and Prem Natarajan. Discriminative Sample Selection for Statistical Machine Translation. In *Proc. EMNLP*, pages 626–635, 2010b.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. ICLR*, pages 1–15, 2015.

Michael Bloodgood and Chris Callison-Burch. Bucking the Trend: Large-Scale Cost-Focused Active Learning for Statistical Machine Translation. In *Proc. ACL*, pages 854–864, 2010.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In *Proc. EMNLP*, pages 858–867, 2007.

Peter F. Brown, Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19:263–312, 1993.

Jean-Cédric Chappelier, Martin Rajman, et al. A Generalized CYK Algorithm for Parsing Stochastic CFG. In *Proc. TAPD*, volume 98, pages 133–137. Citeseer, 1998.

Stanley F. Chen and Joshua Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proc. ACL*, pages 310–318, 1996.

David Chiang. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228, 2007.

Trevor Cohn and Mirella Lapata. Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora. In *Proc. ACL*, pages 728–735, 2007.

Raj Dabre, Fabien Cromieres, Sadao Kurohashi, and Pushpak Bhattacharyya. Leveraging Small Multilingual Corpora for SMT Using Many Pivot Languages. In *Proc. NAACL*, pages 1192–1202, 2015.

Adrià de Gispert and José B. Mariño. Catalan-English Statistical Machine Translation without Parallel Corpus: Bridging through Spanish. In *Proc. of LREC 5th Workshop on Strategies for developing machine translation for minority languages*, pages 65–68, 2006.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-Task Learning for Multiple Language Translation. In *Proc. ACL*, pages 1723–1732, 2015.

Chris Dyer, Aaron Cordova, Alex Mont, and Jimmy Lin. Fast, Easy, and Cheap: Construction of Statistical Machine Translation Models with MapReduce. In *Proc. WMT*, pages 199–207, 2008.

Matthias Eck, Stephan Vogel, and Alex Waibel. Low Cost Portability for Statistical Machine Translation based in N-gram Frequency and TF-IDF. In *Proc. IWSLT*, pages 61–67, 2005.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. Zero-Resource Translation with Multi-Lingual Neural Machine Translation. In *Proc. EMNLP*, pages 268–277, 2016.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a Translation Rule? In *Proc. NAACL*, pages 273–280, 2004.

Ulrich Germann. Building a statistical machine translation system from scratch: how much bang for the buck can we expect? In *Proc. of the workshop on Data-driven methods in machine translation-Volume 14*, pages 1–8, 2001.

Ulrich Germann. Dynamic phrase tables for machine translation in an interactive post-editing scenario. In *Proc. AMTA 2014 Workshop on Interactive and Adaptive Machine Translation*, pages 20–31, 2014.

Jesús González-Rubio, Daniel Ortiz-Martínez, and Francisco Casacuberta. Active learning for interactive machine translation. In *Proc. EACL*, pages 245–254, 2012.

Isao Goto, Masao Utiyama, Eiichiro Sumita, Akihiro Tamura, and Sadao Kurohashi. Distortion Model Considering Rich Context for Statistical Machine Translation. In *Proc. ACL*, pages 155–165, 2013.

Jonathan Graehl and Kevin Knight. Training Tree Transducers. In *Proc. NAACL*, pages 105–112, 2004.

Spence Green, Sida I. Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. Human Effort and Machine Learnability in Computer Aided Translation. In *Proc. EMNLP*, pages 1225–1236, 2014.

Gholamreza Haffari and Anoop Sarkar. Active Learning for Multilingual Statistical Machine Translation. In *Proc. ACL*, pages 181–189, 2009.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. Active Learning for Statistical Phrase-based Machine Translation. In *Proc. NAACL*, pages 415–423, 2009.

Kenneth Heafield. KenLM: Faster and Smaller Language Model Queries. In *Proc, WMT*, pages 187–197, 2011.

Eduard Hovy. Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. In *Proc. LREC*, pages 535–542, 1998.

Ann Irvine and Chris Callison-Burch. Combining Bilingual and Comparable Corpora for Low Resource Machine Translation. In *Proc. of the Eighth Workshop on Statistical Machine Translation*, pages 262–270, August 2013.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernand a Vi ̄ta gas, Martin Wattenberg, Greg Corrado,

Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *TACL*, 5:339–351, 2017.

Toru Kasai, Gunho Lee, Hiroki Arimura, Setsuo Arikawa, and Kunsoo Park. Linear-Time Longest-Common-Prefix Computation in Suffix Arrays and Its Applications. In *Proc. CPM*, pages 181–192, 2001.

Philip N. Klein. Computing the Edit-Distance Between Unrooted Ordered Trees. In *Proc. of European Symposium on Algorithms*, pages 91–102, 1998.

Philipp Koehn. Statistical Significance Tests for Machine Translation Evaluation. In Dekang Lin and Dekai Wu, editors, *Proc. EMNLP*, pages 388–395, July 2004.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. ACL*, pages 177–180, 2007.

Phillip Koehn, Franz Josef Och, and Daniel Marcu. Statistical Phrase-Based Translation. In *Proc. NAACL*, pages 48–54, 2003.

Tomer Levinboim and David Chiang. Supervised Phrase Table Triangulation with Neural Word Embeddings for Low-Resource Languages. In *Proc. EMNLP*, pages 1079–1083, 2015.

Adam Lopez and Matt Post. Beyond bitext: Five open problems in machine translation. In *Proc. of the EMNLP Workshop on Twenty Years of Bitext*, 2013.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proc. EMNLP*, pages 1412–1421, 2015.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Proc. NAACL*, pages 746–751, 2013.

Akiva Miura, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Improving Pivot Translation by Remembering the Pivot. In *Proc. ACL*, pages 573–577, 2015.

Akiva Miura, Graham Neubig, Michael Paul, and Satoshi Nakamura. Selecting Syntactic, Non-redundant Segments in Active Learning for Machine Translation. In *Proc. NAACL*, pages 20–29, 2016.

Akiva Miura, Graham Neubig, Katsuhito Sudoh, and Satoshi Nakamura. Tree as a Pivot: Syntactic Matching Methods in Pivot Translation. In *Proc. WMT*, pages 90–98, September 2017.

Robert Munro. Crowdsourcing and the crisis-affected community. *Information Retrieval*, 16(2):210–266, 2013.

Makoto Nagao. A framework of a Mechanical Translation between Japanese and English by Analogy Principle. In *Proc. International NATO Symposium on Artificial and Human Intelligence*, pages 173–180, 1984. ISBN 0-444-86545-4.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. ASPEC: Asian Scientific Paper Excerpt Corpus. In *Proc. LREC*, pages 2204–2208, 2016.

Graham Neubig. Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers. In *Proc. ACL Demo Track*, pages 91–96, 2013.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In *Proc. ACL*, pages 529–533, 2011.

Graham Neubig, Philip Arthur, and Kevin Duh. Multi-Target Machine Translation with Multi-Synchronous Context-free Grammars. In *Proc. NAACL*, pages 484–491, 2015.

Sergei Nirenburg. Knowledge-based machine translation. *Machine Translation*, 4(1):5–24, 1989.

Franz Josef Och. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. ACL*, pages 160–167, 2003.

Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Ckylark: A More Robust PCFG-LA Parser. In *Proc. NAACL*, pages 41–45, 2015.

Daisuke Okanohara and Jun'ichi Tsujii. Text Categorization with All Substring Features. In *Proc. SDM*, pages 838–846, 2009.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. ACL*, pages 311–318, 2002.

Michael Paul, Hirofumi Yamamoto, Eiichiro Sumita, and Satoshi Nakamura. On the Importance of Pivot Language Selection for Statistical Machine Translation. In *Proc. NAACL*, pages 221–224, 2009.

Philip Resnik and Noah A Smith. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380, 2003.

Burr Settles and Mark Craven. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *Proc. EMNLP*, pages 1070–1079, 2008.

Claude E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. A Shared Task on Multimodal Machine Translation and Crosslingual Image Description. In *WMT*, pages 543–553, 2016.

Matthias Sperber, Mirjam Simantzik, Graham Neubig, Satoshi Nakamura, and Alex Waibel. Segmentation for Efficient Supervised Language Annotation with an Explicit Cost-Utility Tradeoff. *TACL*, 2:169–180, 2014.

Andreas Stolcke. SRILM - an extensible language modeling toolkit. In *Proc. ICSLP*, pages 901–904, 2002.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

Jörg Tiedemann. News from OPUS-A collection of multilingual parallel corpora with tools and interfaces. In *Proc. RANLP*, volume 5, pages 237–248, 2009.

Katrin Tomanek and Udo Hahn. Semi-Supervised Active Learning for Sequence Labeling. In *Proc. ACL*, pages 1039–1047, 2009.

Marco Turchi, Tijl De Bie, and Nello Cristianini. Learning performance of a machine translation system: a statistical and computational analysis. In *Proc. WMT*, pages 35–43, 2008.

Masao Utiyama and Hitoshi Isahara. A Comparison of Pivot Methods for Phrase-Based Statistical Machine Translation. In *Proc. NAACL*, pages 484–491, 2007.

Bernard Vauquois. A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Mechanical Translation. In *Proc. IFIP Congress (2)*, volume 68, pages 1114–1122, 1968.

Katharina Wäschle and Stefan Riezler. Analyzing Parallelism and Domain Similarities in the MAREC Patent Corpus. *Multidisciplinary Information Retrieval*, pages 12–27, 2012.

Mikio Yamamoto and Kenneth Ward Church. Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus. *Computational Linguistics*, 27(1):1–30, 2001.

Omar F Zaidan and Chris Callison-Burch. Crowdsourcing translation: Professional quality from non-professionals. In *Proc. ACL*, pages 1220–1229, 2011.

Xiaoning Zhu, Zhongjun He, Hua Wu, Conghui Zhu, Haifeng Wang, and Tiejun Zhao. Improving Pivot-Based Statistical Machine Translation by Pivoting the Co-occurrence Count of Phrase Pairs. In *Proc. EMNLP*, pages 1665–1675, 2014.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The United Nations Parallel Corpus v1.0. In *Proc. LREC*, pages 3530–3534, 2016.

Barret Zoph and Kevin Knight. Multi-Source Neural Translation. In *Proc. NAACL*, pages 30–34, June 2016.

Publication List

Refereed Domestic Journal Papers

1. 三浦明波, Graham Neubig, Sakriani Sakti, 戸田智基, 中村哲. 中間言語情報を記憶するピボット翻訳手法. 自然言語処理, Vol.23, No.5, pp499-528, December 2016.

2. 三浦明波, Graham Neubig, Michael Paul, 中村哲. 統語的一貫性と非冗長性を重視した機械翻訳のための能動学習手法. 自然言語処理, Vol.24 No.3, pp463-489, June 2017.

Refereed International Conference Papers

1. Akiva Miura, Graham Neubig, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura. Improving Pivot Translation by Remembering the Pivot. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), July 2015.

2. Akiva Miura, Graham Neubig, Michael Paul, Satoshi Nakamura. Selecting Syntactic, Non-redundant Segments in Active Learning for Machine Translation. Proceedings of the 15th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL), June 2016.

3. Akiva Miura, Graham Neubig, Katsuhito Sudoh, Satoshi Nakamura. Tree as a Pivot: Syntactic Matching Methods in Pivot Translation. Proceedings of the Second Conference on Machine Translation (WMT), September 2017.

International Workshop Papers

1. Raphael Shu, Akiva Miura. Residual Stacking of RNNs for Neural Machine Translation. Proceedings of the 3rd Workshop on Asian Translation (WAT), December 2016.

Domestic Conference Papers

1. 三浦明波, Graham Neubig, Sakriani Sakti, 戸田智基, 中村哲. 階層的フレーズベース翻訳におけるピボット翻訳手法の応用. 情報処理学会第 219 回自然言語処理研究会 (SIG-NL), December 2014.

2. 三浦明波, Graham Neubig, Sakriani Sakti, 戸田智基, 中村哲. 中間言語モデルを用いたピボット翻訳の精度向上. 情報処理学会第 222 回自然言語処理研究会 (SIG-NL), July 2015.

3. 三浦明波, Graham Neubig, Michael Paul, 中村哲. 構文木と句の極大性に基づく機械翻訳のための能動学習. 情報処理学会第 224 回自然言語処理研究会 (SIG-NL), December 2015.

4. 三浦明波, Graham Neubig, Michael Paul, 中村哲. 構文情報に基づく機械翻訳のための能動学習手法と人手翻訳による評価. 言語処理学会第 22 回年次大会 (NLP2016), March 2016.

5. 三浦明波, Graham Neubig, 中村哲. 木構造を中間表現とするピボット翻訳手法. 情報処理学会第 227 回自然言語処理研究会 (SIG-NL), July 2016.

Awards

1. Asia-Pacific Association for Machine Translation (AAMT), Nagao Award Student Award, June 2016.

2. NAIST Top Scholarship Program for Excellent Academic Standing, July 2016.

Master's Thesis

1. 三浦明波. 中間言語モデルを用いた多言語機械翻訳の精度向上. Master's thesis, Graduate School of Information Science, Nara Institute of Science and Technology, March 2016.