A Japanese-to-English Statistical Machine Translation System for Technical Documents

Katsuhito Sudoh

Abstract

This thesis addresses a Japanese-to-English statistical machine translation (SMT) system for technical documents. Machine translation (MT) is a promising solution for growing translation needs. Japanese-to-English MT is one of the most difficult language pairs due to their large lexical and syntactic differences. This thesis work focuses on patents as the most demanded technical documents that have different attributes from other general documents: technical terms and long complex sentences. This thesis tackles three important research problems in the target task: word segmentation on technical terms, unknown katakana word transliteration, and long-distance reordering. Novel techniques are proposed to overcome these problems: domain adaptation of word segmentation using very large-scale patent data, noise-aware translation fragment extraction for accurate machine transliteration, and syntax-based post-ordering for efficient and accurate long-distance reordering.

Chapter 2 gives a brief introduction of SMT techniques on which the proposed methods are based. They are established and widely used in various language pairs but not sufficient for the Japanese-to-English patent SMT.

Chapter 3 presents a novel domain adaptation method for the Japanese word segmentation, using very large-scale Japanese monolingual unlabeled corpora. The proposed method utilizes word boundary clues called Branching Entropy and pseudo-dictionary features obtained from the Japanese monolingual corpora. The probabilistic characteristic of the Branching Entropy mitigates the stability issue of the baseline method using Accessor Variety. The method achieved word segmentation F-measure of 98.36% and out-of-vocabulary word recall of 92.61% in word segmentation experiments, which were significantly higher than the performance of the baseline methods.

Chapter 4 presents a novel noise-aware character alignment method which extracts

meaningful transliteration fragments. Although more than a half of unknown words in Japanese-to-English patent SMT are katakana words, they can be translated into the original English words. However, the transliteration is not straightforward because of the ambiguous and inconsistent mapping between katakana and English phonemes. This work focuses on partial noise in transliteration candidates extracted from the bilingual corpora to learn the mapping, which has not been addressed by previous studies that model sample-wise noise only. The proposed method achieved transliteration accuracy of 66% for unknown katakana words, which is 10% error reduction from the method addressing sample-wise noise only.

Chapter 5 presents a novel efficient SMT method called post-ordering that divides the SMT problem explicitly into two steps: monotone lexical translation by the phrase-based SMT and reordering by the syntax-based SMT. The post-ordering approximates the accurate but computationally expensive syntax-based SMT by using an intermediate language with English words in the Japanese word order. The post-ordering achieved accurate translation comparable to the syntax-based SMT with more than six-time faster decoding speed.

Chapter 6 presents a patent SMT system integrating the techniques presented above. This system has two major advantages other than the advantages of the individual techniques. First, domain adaptation of Japanese pre-processing is needed only on word segmentation, not on more difficult Japanese parsing. Second, katakana unknown words are translated prior to reordering and expected to be reordered correctly without special treatment of unknown word reordering. The system achieved the BLEU scores of 34.77% and 35.75% for the NTCIR-9 and NTCIR-10 PatentMT test sets, which were consistently higher than the performance of the baseline systems using the standard techniques.

Chapter 7 concludes the thesis. The proposed SMT framework realizes a practical Japanese-to-English SMT system adapted to technical documents, where many technical terms and long sentences cause serious translation errors. The propsoed methods do not rely on additional human annotations on in-domain corpora, and can be trained with existing bilingual and monolingual corpora. Finally, some further prospects are discussed.

Acknowledgments

First and foremost, I would like to thank Professor Tatsuya Kawahara for his supervision. His thoughtful advice for this thesis work encouraged me greatly. My doctor course period in his lab. was a great opportunity for different experiences, not only in my research work.

I would like to thank Professor Sadao Kurohashi and Professor Hisashi Kashima for serving on my doctoral committee. Their advice was very helpful for improving this thesis.

I would also like to thank Professor Shinsuke Mori for fruitful discussion with him. His expertise in adaptable natural language processing helped this thesis work largely.

I also appreciate Dr. Masaaki Nagata for his leading my machine translation research at NTT and various supports for this thesis work. His deep insight on machine translation and natural language processing helped this thesis work. I would like to thank Mr. Hajime Tsukada and Prof. Hideki Isozaki for their kindly help in everything on my work at NTT. They guided me to a really exciting research field of machine translation.

I would like to express my thanks to my colleagues at NTT. Dr. Kevin Duh and Dr. Xianchao Wu gave me a lot of funs in my work by their impressive works and interesting discussions. The time with them was the most exciting period in my work life so far. Dr. Taro Watanabe taught me many things about statistical machine translation, which motivated me greatly. Dr. Tsutomu Hirao and Dr. Jun Suzuki gave me valuable advice on this thesis work. Many other people who are working or worked together also helped me.

I would like to thank the members of Kawahara lab., especially Dr. Yuya Akita and Dr. Koichiro Yoshino for their various support on the time in lab. It was a really fun for me to come to the lab. and talk with them.

Finally, thank you Emiko and Vanilla, for always being there with me.

CONTENTS

Contents

List of Tables

List of Figures

Chapter 1

Introduction

There are a large number of languages all over the world, which have evolved differently according to their historical and cultural backgrounds. *Translation* has been the most fundamental tool for human communication between those using different languages. Translation is a very difficult problem in general, due to language differences and large ambiguity of linguistic expressions. For that reason, translation requires expert language skills in both languages to be translated from (called *source* language) and into (called *target* language). Nowadays there are strong needs of the translation not only in international business and diplomacy but also in daily life — for travel, social network, education, and so on. These needs will grow more rapidly than before along with globalization of the society. Machine translation (MT) is a highly demanded technology for these growing needs that cannot be satisfied only by the human-intensive translation by expert translators in terms of the cost and quickness.

MT is a very challenging problem because of the difficulty of translation even by humans. The history of MT research and development for more than sixty years has suffered from the difficulty, but it has also driven various studies in computational linguistics and natural language processing and development of many language resources including treebanks, bilingual dictionaries, and corpora. Thanks to these long-time efforts, MT is now used in practice for some relatively easy translation tasks: translation between similar languages such as English-French and Japanese-Korean, and translation of simple or typical sentences in limited situations. For more difficult translation tasks with other language

pairs and broader domains, MT is still a difficult problem and needs more improvement for a practical use.

This thesis work is to tackle problems in Japanese-to-English statistical MT (SMT) for technical documents.

1.1 Target Task of Machine Translation

The translation tasks can be characterized roughly by following aspects:

- language pairs (source and target languages),
- purposes,
- types of documents to be translated, and
- translation (MT in this work) approaches.

This section discusses the target task, Japanese-to-English SMT for technical documents, in terms of these aspects.

1.1.1 Language Pair of Machine Translation

The language pair is a fundamental aspect of the problem of translation not only MT. In human translation, translators should have sufficient language skills both on source and target languages. Considering large differences among languages, translation of different language pairs requires different skills and experiences for translators. This is also the case for MT; MT must be set up and tuned differently for different language pairs, such as dictionaries, syntactic parsers, and various kinds of system parameters.

The problem on the language pair depends on the extent of language differences. There are a variety of languages; some of them are very similar, and some of them are very different even if they are used in geographically close areas. Translation for such different language pairs is obviously difficult for both human translation and MT. There are roughly two different kinds of language differences: lexical and syntactic gaps. The lexical gap can be bridged by dictionary information, but lexicons of different languages do not match one-to-one in general and their complete mappings cannot be obtained easily. The syntactic gap

can be bridged by transformation of syntactic structures, such as from Subject-Verb-Object to Subject-Object-Verb, and from modifier-modifee to modifee-modifier. These transformations are usually complex and not trivial.

For the Japanese people, MT between Japanese and English is one of the most important language pairs. It is also important for non-Japanese speakers to understand Japanese documents and speeches. However, the performance of Japanese-English MT is poor except for some easy situations, because of their large lexical and syntactic gaps. Japanese and English lexicons are very different; meanings of Japanese words in English cannot be predicted from their surface forms, different from other Western languages that share many Latin-origin words with English. In a syntactic view, Japanese is a head-final, Subject-Object-Verb language, different from English structure of basically head-initial, Subject-Verb-Object language. This thesis discusses how to bridge these gaps in Japanese-to-English MT. There are also important needs for MT in the opposite direction — English-to-Japanese. English-to-Japanese MT has evolved largely in recent several years based on the Japanese head-final syntax. On the other hand, Japanese-to-English MT is still difficult and worse than the English-to-Japanese MT due to asymmetry between translation directions.

1.1.2 Purpose of Machine Translation

One typical and important purpose of translation is to read foreign language documents such as books, newspapers, and webpages. There are various purposes of translation, which can be classified roughly into the followings:

Assimilation Translating foreign language documents into a language demanded by users for understanding their content.

Dissemination Translating documents into different languages for users to provide translated documents.

Communication Translating conversations in different languages for users to understand each other.

These have different requirements according to their use cases. In the communication translation, these are two important attributes, bi-directionality and translation speed, for

real-time cross-lingual communication. The assimilation and dissemination MT do not always require bi-directionality, because they are motivated for providing translations for written and spoken documents. The speed issue also does not matter so strictly except for some on-line translation tasks such as simultaneous interpretation. One important problem in the assimilation and dissemination MT is the amount of documents to be translated, which does not matter in the communication MT. Translation of stable documents can easily be parallelized in MT, so MT is suitable for translating a large amount of stable documents.

The target task in this thesis work is for the assimilation and dissemination to provide English translations of Japanese technical documents. Remarkable growth of publicly available language information in these years increases the importance of this task for obtaining and distributing information in different languages. There are a large number of Japanese documents in Japan, in which a very limited portions have been translated so far.

1.1.3 Document Type of Machine Translation

There are various types of documents to be translated, written and spoken, descriptive and rhetorical, short and long, and so on. These documents have different attributes with respect to translation difficulties. This thesis work focuses on two attributes: syntactic and semantic complexities. The syntactic complexity is roughly proportional to the sentence length, and the semantic complexity is affected by the use of rhetorical and non-literal linguistic expressions. Figure 1.1 shows a brief mapping for several types of documents in terms of the semantic complexity (X-axis) and the syntactic complexity (Y-axis). Newspaper articles often have long and complex sentences but usually use literal expressions because newspapers are to present specific facts. Technical documents such as scientific papers, manuals, and patents contain long sentences but focus on presenting facts in rigid expressions. Daily conversations are more casual and shorter than written documents but often include well-known non-literal expressions such as idioms. On the other hand, poetry and novels include many rhetorical expressions that cannot be literally translated into other languages. The semantic complexity largely affects the difficulty of translation, even for human translators. Casual languages used in informal conversations and social networks are also difficult to translate because they have little literal correspondences across differ-

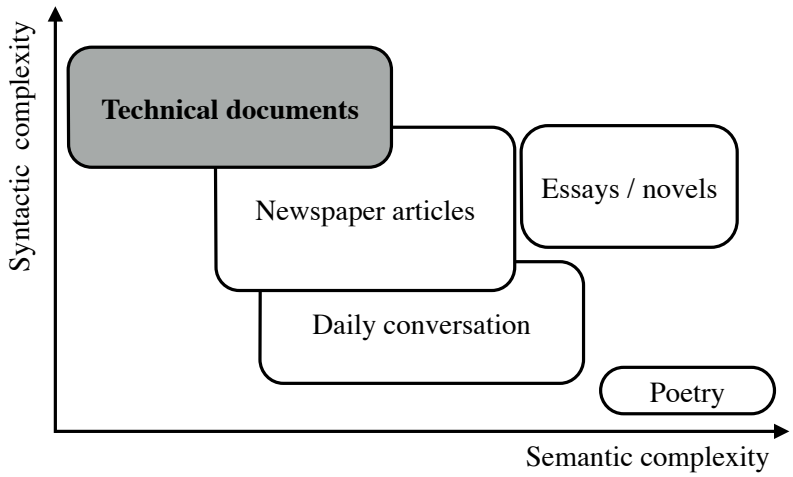


Figure 1.1: Relation between semantic and syntactic complexities for different document types

ent languages. Such documents need deep semantic understanding and insightful language generation that are too difficult even by recent natural language technologies.

MT is expected to work well in literal translation without rhetorical techniques, because MT generates translations by composing partial translations for various language expressions based on the principle of compositionality. From this viewpoint, technical documents can be translated literally in general and are suitable for MT. The technical documents have large difficulty in translating their syntactically complex sentences while they can be translated literally. There is an important technical challenge to translate long and complex sentences accurately and efficiently, because there are severe problems both in modeling and search in MT.

This thesis work focuses on patents among various kinds of technical documents, which are one of the most important MT tasks for the following reasons. First, patents contain novel technical information about inventions that can be used widely in the world. Second, they are open to public but basically domestic documents written in official languages of the countries to which they are filed, so they have to be translated into other languages. Finally, they are typical well-organized, descriptive written documents suitable for MT.

1.1.4 Machine Translation Approach

In the long history of the research and development of MT, several approaches has evolved by various technologies and language resources. An early approach to MT was the rule-based (or knowledge-based) MT (RBMT) by using bilingual dictionaries and hand-crafted translation pattern rules. These bilingual resources are carefully developed by experts with sufficient knowledge of the source and target languages. The performance of the RBMT depends on the quality of the bilingual resources, and is practically good in some language pairs and domains as used in some commercial products. The RBMT is reasonable and straightforward but needs efforts by expert developers with high proficiency to cover numerous language expressions without rule conflicts. It becomes further problematic in the development of the RBMT for different languages and domains, which often requires very different pattern rules that should be hand-crafted from scratch.

In later years, the corpus-based approach was proposed and has been investigated up to the present time, together with continuous effort for developing multilingual corpora of various languages and domains. One of the approaches is the example-based MT (EBMT), which composes translations using existing translation examples in the corpus. One important advantage of the EBMT is that it works effectively for source language sentences that are very similar to examples in the corpus. Such sentences can be translated easily by small edits on matched examples, even if the sentences are very long and complex. The actual performance of the EBMT is highly dependent on the corpus size but the large corpus also raises problems of large-scale example retrieval and conflict resolution. Another corpus-based approach is the statistical MT (SMT) that represents the translation process by statistical transductions of sentence components (words and phrases) using statistical models learned from the bilingual corpus. It gives a reasonable solution for the conflict problem in the RBMT and EBMT by probabilistic evidences. It has rapidly evolved with the growth of statistical natural language processing and machine learning techniques, from its early establishment on the late 1980s. The most important advantage of the corpus-based MT is its rapid deployment for given corpora, basically independent from its source and target languages and domains. This is a practically beneficial advantage against the hand-crafted RBMT.

Among many kinds of technical documents, there are a large number of patents filed so far and many of them have been translated and also filed in other countries. These translated patents can be used as bilingual corpora for training SMT models. The target task is a good practical use case of the SMT that can be trained using large-scale documents for translating not-yet-translated and future documents.

1.2 Problems

The focus of this thesis work is Japanese-to-English SMT for patents. Patents are often translated into different languages to be filed to several countries and they are associated each other (a patent family), so that we can obtain large-scale multilingual patent corpora for the use in the patent SMT (Utiyama and Isahara, 2007). There are patent translation shared tasks for research on the patent MT between Japanese and English (Fujii et al., 2008; Fujii et al., 2010; Goto et al., 2011; Goto et al., 2013), and many MT methods had been studied and applied to them.

One important problem is a lexical (or morphological) problem, appearance of many technical terms in patents of various technical fields. These technical terms are often unknown words in two different components in the patent SMT: Japanese word segmentation and the SMT itself. Since Japanese orthography does not have explicit word boundaries, word segmentation is a fundamental pre-processing in the Japanese-to-English SMT. Not a few technical terms cannot be segmented correctly by existing word segmenters trained with general documents such as newspapers. The incorrectly segmented words will be translated incorrectly by the SMT. Even if technical terms are segmented correctly, they may be unknown for the SMT and cannot be translated, due to the lack of bilingual correspondence in the training corpora. The technical terms usually hold important information in patents, and should be dealt carefully in the patent MT. Typically technical terms written in katakana phonograms are often unknown in the Japanese-to-English SMT, because many different English names and concepts are imported into Japanese by transliterating them into katakana. Since many katakana words are not translated as unknown words in a Japanese-to-English patent SMT, translation of the katakana unknown words is important. These katakana words can be back-transliterated into English by using some phonetical

mappings even if their actual word-based translations cannot be obtained from existing bilingual corpora. In summary, the following two research problems arise:

(1) word segmentation of Japanese unknown technical terms, and
(2) back-transliteration of unknown katakana words.

Another major problem is a syntactic problem. The large difference in the word order between Japanese and English becomes very severe in long sentences in patents. Since the SMT tries to find translation hypotheses as a search problem over numerous number of different word translations and word order, its computational complexity increases very rapidly with the lengths of the input sentence. The search in the word order, usually called *reordering*, has the complexity of $O(n!)$ for the input length n in theory by a naive permutation-based search. It can be reduced by limiting reordering distance, but it is not suitable for the patent SMT between Japanese and English that requires long distance reordering. Thus, the following reseach problem arises:

(3) long-distance reordering for long patent sentences with acceptable computation time.

The three research problems above are important especially in the Japanese-to-English patent SMT. The goal of this thesis work is to improve Japanese-to-English patent SMT by solving these problems. Here is an underlying issue in the SMT for documents in specific domains. Most of sophisticated language resources such as bilingual dictionaries and treebanks are usually developed on general-domain data, typically newspaper articles. Although the problems above may be mitigated by using term dictionaries and other patent-oriented language resources, the development of such resources requires careful human-intensive work and is very costly. This thesis work takes this issue into account.

1.3 Approach

The approach of this thesis work to these problems is based on semi-supervised or unsupervised learning techniques without any specially developed dictionaries, treebanks, and other annotated language resources for the patent documents, in order to be applied with minimal human-intensive efforts.

For the word segmentation problem, this thesis work uses very large-scale Japanese patent corpora to adapt Japanese word segmenter to the patent domain by a semi-supervised learning framework. Although some technical terms do not appear in general-domain word-segmented corpora that are commonly used for training word segmenters, they are expected to appear many times in the patent documents. Very large-scale unlabeled corpora without correct word segmentation information is used for the domain adaptation, based on an intuition that reliable word segmentation clues can be derived even from such unlabeled corpora. This approach has been investigated previously, but this work extends the approach with more stable word segmentation clues for the use with the very large-scale patent corpora.

For the transliteration problem, this thesis work uses a statistical transliteration technique for the katakana unknown words. The transliteration is regarded as a character-based SMT, whose models are trained using a character-based parallel corpus. This thesis work obtains a character-based training corpus from the sentence-based patent bilingual corpora to learn transliterations used in the patent domain by an unsupervised method without hand-crafted transliteration dictionaries. A novel noise-aware alignment method for extracting noise-free transliteration fragments is proposed to tackle a partial noise problem in this character-based training corpus.

For the long distance reordering problem, this thesis work proposes a novel SMT method called *post-ordering*. In this approach, Japanese sentences are translated into Japanese-ordered English by monotone lexical SMT and then the Japanese-ordered English sentences are reordered into correct-ordered English by syntax-based SMT. This two-stage translation largely reduces the computational complexity of the original joint problem of lexical translation and reordering. The method enables efficient long distance reordering without performance degradation in translation accuracy. Since the intermediate Japanese-ordered English are generated easily from English sentences based on the Japanese head-final syntax, translation models for the two translation tasks are trained from Japanese-English bilingual corpora.

Finally, a Japanese-to-English patent SMT system is built by integrating these techniques, whose architecture is shown in Figure 1.2.

1.4 Organization of This Thesis

The organization of this thesis is as follows. Chapter 2 gives a brief review of the SMT: fundamental formulation, models, decoding, and evaluation. It covers basic techniques used in this thesis work. Chapter 3 presents the patent-adapted Japanese word segmentation. The word segmentation is an important preprocessing for most of Japanese language processing, not only for the Japanese-to-English SMT. This work focuses on its domain adaptation to the patent domain. Chapter 4 presents the statistical transliteration for translating unknown technical terms written in katakana. Its model is trained using a sentence-level parallel corpus, not using a word-level dictionary of transliterated words. Chapter 5 presents an efficient syntax-based translation with the post-ordering framework. It divides lexical translation and reordering explicitly to reduce the large computational complexity of accurate syntax-based reordering. Chapter 6 presents an overall Japanese-to-English patent MT system using the proposed techniques with the architecture shown in Figure 1.2. Chapter 7 concludes this thesis and gives some future directions of the Japanese-to-English SMT.

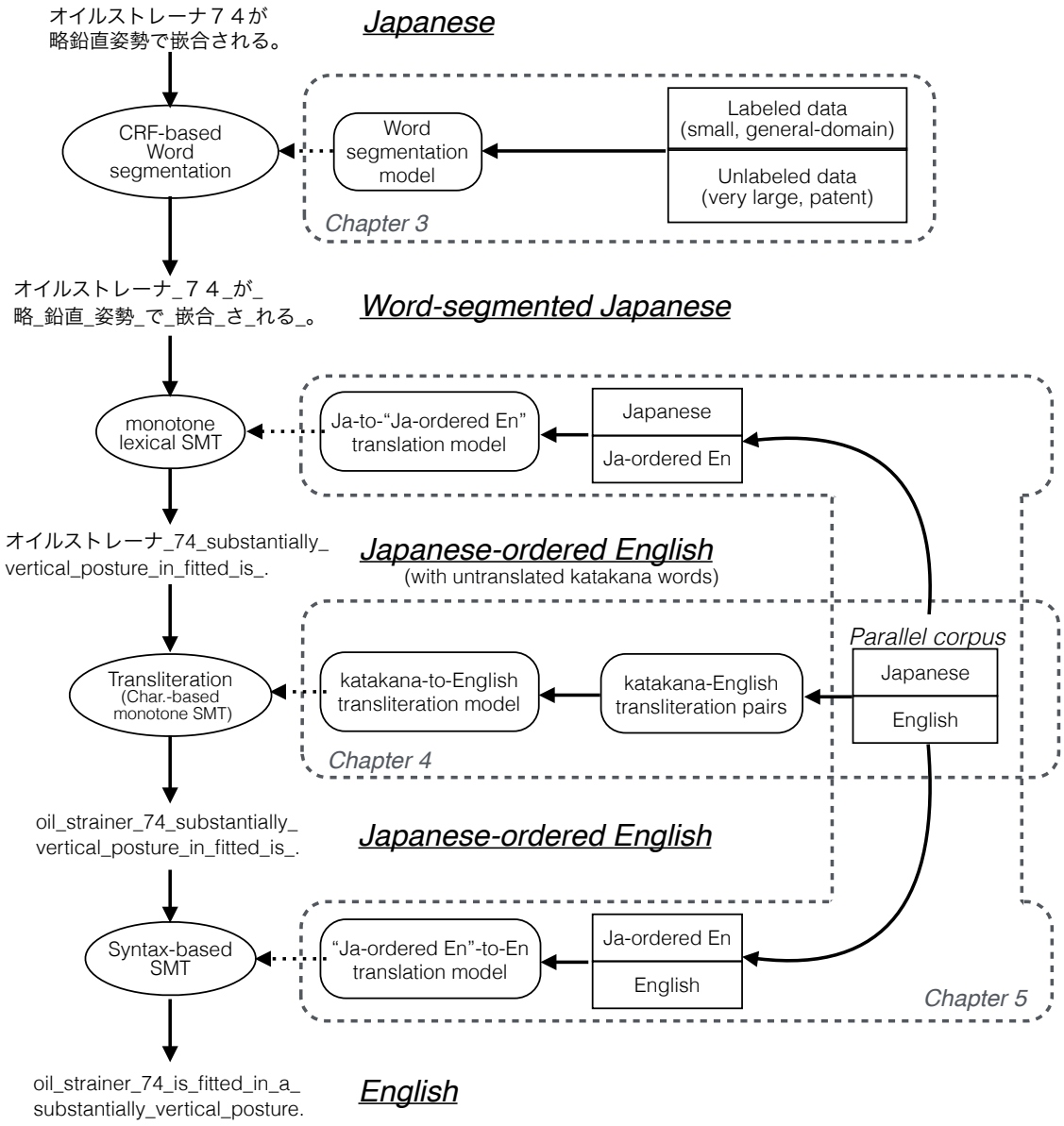


Figure 1.2: Architecture of the system presented in this thesis

Chapter 2

Statistical Machine Translation

2.1 Fundamentals

2.1.1 SMT Based on Noisy Channel Model

The fundamental idea of the SMT was established by Brown et al. (1993), based on a noisy channel model. Suppose a French sentence $\boldsymbol{f}$ is translated into a English $\boldsymbol{e}$. French and English are called *source* and *target* languages of this translation process. Here this translation process can be modeled by a noisy channel model in which a English sentence $\boldsymbol{e}$ is *encoded* into a French sentence $\boldsymbol{f}$. The translation problem can be seen as a *decoding* problem from $\boldsymbol{f}$ to $\boldsymbol{e}$, in which it aims to find the best English sentence $\hat{\boldsymbol{e}}$ that maximizes the posterior probability of the English sentence given a French sentence $\boldsymbol{f}$:

$$\hat{\boldsymbol{e}} = \underset{\boldsymbol{e} \in E(\boldsymbol{f})}{\arg\max}\, p(\boldsymbol{e}|\boldsymbol{f})\,, \tag{2.1}$$

where $E(\boldsymbol{f})$ is a set of all possible English translations from $\boldsymbol{f}$. The posterior probability $p(\boldsymbol{e}|\boldsymbol{f})$ can be converted as follows by the Bayes' theorem:

$$\hat{\boldsymbol{e}} = \underset{\boldsymbol{e} \in E(\boldsymbol{f})}{\arg\max}\, \frac{p(\boldsymbol{f}|\boldsymbol{e})\, p(\boldsymbol{e})}{p(\boldsymbol{f})} = \underset{\boldsymbol{e} \in E(\boldsymbol{f})}{\arg\max}\, p(\boldsymbol{f}|\boldsymbol{e})\, p(\boldsymbol{e})\,, \tag{2.2}$$

because the probability of the given French sentence $\boldsymbol{f}$ is a constant. The original posterior probability $p(\boldsymbol{e}|\boldsymbol{f})$ is decomposed into two different probabilities:

- Language model for computing probability $p(\boldsymbol{e})$: A model of English sentences that gives high probability to fluent ones, and

- Translation model for computing probability $p(\boldsymbol{f}|\boldsymbol{e})$: A model of French sentences according to given English sentences that gives high probability to French sentences corresponding well with the given English sentences.

2.1.2 SMT Based on Log-Linear Models

Och and Ney (2002) proposed the use of a discriminative framework instead of the Bayes decision rule in Equation (2.2), as represented by the following decision rule:

$$\hat{\boldsymbol{e}} = \underset{\boldsymbol{e} \in E(\boldsymbol{f})}{\arg\max} \frac{\exp\left(\sum_m \lambda_m h_m\left(\boldsymbol{f}, \boldsymbol{e}\right)\right)}{\sum_{\boldsymbol{e'} \in E(\boldsymbol{f})} \exp\left(\sum_m \lambda_m h_m\left(\boldsymbol{f}, \boldsymbol{e'}\right)\right)} = \underset{\boldsymbol{e} \in E(\boldsymbol{f})}{\arg\max} \sum_m \lambda_m h_m\left(\boldsymbol{f}, \boldsymbol{e}\right), \qquad (2.3)$$

where $h_m\left(\boldsymbol{f}, \boldsymbol{e}\right)$ is m-th feature function defined on the given French sentence $\boldsymbol{f}$ and a translation hypothesis $\boldsymbol{e}$. This discriminative framework enables more flexible use of various features such as the length of $\boldsymbol{e}$, a reverse-direction translation model $p(\boldsymbol{e}|\boldsymbol{f})$, the number of French and English words included in a bilingual dictionary, in addition to the language and translation models used in the noisy channel framework. The Bayes decision rule in Equation (2.2) is a special case in this framework with two equal-weight feature functions of translation and language model probabilities.

Recent advances in the field of SMT were derived from this discriminative approach. One important progress in this approach is the use of automatic evaluation metrics (described later in section 2.4) for optimizing parameters λ_m. Och (2003) proposed a minimum error rate training (MERT) method to optimize the parameters according to an evaluation metric. This enables direct optimization of the SMT towards better translation quality in a certain evaluation metric, while a maximum likelihood optimization (Och and Ney, 2002) does not guarantee the improvement in translation quality.

2.2 Models

2.2.1 Word-based Translation Model

An early attempt for the translation model is a word-based model (Brown et al., 1993). The probability of generating $\boldsymbol{f} = f_1, ..., f_J$ given $\boldsymbol{e} = e_1, ..., e_I$ is defined as the sum of

probabilities of generating $\boldsymbol{f}$ with different *alignments* $\boldsymbol{a}$:

$$p(\boldsymbol{f}|\boldsymbol{e}) = \sum_{\boldsymbol{a}} p(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e}) \,. \tag{2.4}$$

Here, the alignment is a set of correspondences between $\boldsymbol{f}$ and $\boldsymbol{e}$, represented by pairs of word indices like $\{(1,1),(2,3),(3,2)\}$. Some source and target language words may not have their counterparts, so *NULL* alignments are introduced to represent them. Since the correspondences are theoretically many-to-many, the calculation of the probability in Equation (2.4) has a very large computational complexity. Brown et al. (1993) approximated it by considering only one-to-many correspondences from the target language to the source language, in which a source language word is aligned with only one target language word. This approximation largely reduces the computational complexity and helps the model inference to be tractable.

Brown et al. (1993) proposed five different models: Model 1 to 5, which are now called IBM Models. Each of them models $p(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e})$ in a different complexity focusing on reordering, change of word order in translation. Vogel et al. (1996) proposed a different model based on Hidden Markov Models (HMMs) that constrains alignments from adjacent source language words. Parameters of these models can be trained using a parallel corpus, a set of corresponding source and target language sentences. Details of these models and their inference are beyond the scope of this thesis. Refer to chapter 4.1 to 4.4 in the book (Koehn, 2010) for their further details.

The original purpose of these models is the inference of the translation model $p(\boldsymbol{f}|\boldsymbol{e})$ in Equation (2.1). However, they can also be used to find most plausible alignments $\hat{\boldsymbol{a}}$ between source and target language sentences as follows:

$$\hat{\boldsymbol{a}} = \arg\max_{\boldsymbol{a}} p(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e}) \,. \tag{2.5}$$

This is a Viterbi approximation of Equation (2.4). This process is called *word alignment* and its results are used as basic bilingual correspondences for sophisticated SMT methods described next.

2.2.2 Phrase-based Translation Model

The word-based models reduce many-to-many correspondences into one-to-many ones, so some phrasal correspondences cannot be represented. Och et al. (1999) proposed alignment templates induced from the results of the word alignment, and Koehn et al. (2003) extended it to phrase-based MT (PBMT). The key idea of the PBMT is the use of bilingual phrase pairs defined by many-to-many word alignments. Since the word alignment only gives one-to-many correspondences, Koehn et al. (2003) induced phrase pairs as many-to-many correspondences that do not violate both bidirectional word alignment for $p(\boldsymbol{f}|\boldsymbol{e})$ and $p(\boldsymbol{e}|\boldsymbol{f})$, with some heuristics.

The PBMT can be formulated by a noisy channel model similarly to Equation (2.2):

$$\begin{aligned}\hat{\boldsymbol{e}} &= \arg\max_{\boldsymbol{e} \in E(\boldsymbol{f})} p(\boldsymbol{f}|\boldsymbol{e})\, p(\boldsymbol{e}) \\ &= \arg\max_{\boldsymbol{e} \in E(\boldsymbol{f})} \sum_{\boldsymbol{\phi}, \boldsymbol{\alpha}} p(\boldsymbol{f}, \boldsymbol{\phi}, \boldsymbol{\alpha}|\boldsymbol{e})\, p(\boldsymbol{e})\,, \end{aligned} \tag{2.6}$$

where $\boldsymbol{\phi}$ is a sequence of phrase pairs corresponding to the target language sentence $\boldsymbol{e}$, and $\boldsymbol{\alpha}$ is a phrasal alignment. This is an extended reformulation of the word-based translation in Equation (2.4), considering different segmentations into phrases. The phrase pairs in $\boldsymbol{\phi}$ is ordered according to their target language order, and the phrasal alignment $\boldsymbol{\alpha}$ is a set of one-to-one correspondences of the phrase pairs. Suppose $\boldsymbol{f}$ and $\boldsymbol{\alpha}$ is independent from $\boldsymbol{e}$ because the phrase pairs in $\boldsymbol{\phi}$ include the information of $\boldsymbol{e}$, Equation (2.6) can be approximated as follows:

$$\hat{\boldsymbol{e}} \approx \arg\max_{\boldsymbol{e} \in E(\boldsymbol{f})} \sum_{\boldsymbol{\phi}, \boldsymbol{\alpha}} p(\boldsymbol{f}, \boldsymbol{\alpha}|\boldsymbol{\phi})\, p(\boldsymbol{\phi}|\boldsymbol{e})\, p(\boldsymbol{e})\,. \tag{2.7}$$

The translation probability is decomposed into three parts:

- Language model $p(\boldsymbol{e})$: A model of target language sentences,
- Phrase translation model $p(\boldsymbol{\phi}|\boldsymbol{e})$: A model of phrase pair sequences according to given target language sentences, which segments a target language sentence into a sequence of target language phrases and gives their translation into corresponding source language phrases, and

- Phrase reordering model $p(\boldsymbol{f}, \boldsymbol{\alpha}|\boldsymbol{\phi})$: A model of source language sentences considering reordering of phrases.

This PBMT model can also be extended with the log-linear framework described in section 2.1.2, which is the current standard SMT method implemented in the widely-used open source SMT toolkit *Moses*[1] (Koehn et al., 2007).

2.2.3 Syntax-based Translation Model

The PBMT enables to capture phrasal correspondence between the source and target languages, but it segments a sentence into a sequence of continuous non-overlapping phrases. This formulation cannot represent a *gappy* correspondence between "not" in English and "ne ... pas" in French. Furthermore, the phrase reordering model of the PBMT is based on linear ordering and cannot capture hierarchical structures of languages.

Yamada and Knight (2001) proposed a translation model with the target language syntax based on several simple rewriting operations on syntax trees in a noisy channel model. Such an syntax-based approach has been extended to tree-based SMT using syntax of either or both languages (Eisner, 2003; Galley et al., 2004; Liu et al., 2006), based on synchronous tree substitution grammars (STSGs). These methods utilize syntactic structures and labels obtained by syntactic parsers for the source and/or target languages. In contrast, Wu (1997) introduced a formal syntax for bilingual synchronous parsing called inversion transduction grammars (ITGs). This method induces synchronous binary trees with reordering information from a parallel sentence, without using syntactic labels. Chiang (2007) extended this formal syntax-based approach to hierarchical phrase-based MT (HPBMT), based on synchronous context-free grammars (SCFGs) only with two non-terminal symbols "S" (sentence) and "X" (other subtrees). Zollmann and Venugopal (2006) proposed a method that augments the HPBMT approach by using syntactic labels, called syntax-augmented MT (SAMT).

[1] http://www.statmt.org/moses/

The syntax-based MT is formulated as follows:

$$\begin{aligned}\hat{\boldsymbol{e}} &= \arg\max_{\boldsymbol{e}\in E(\boldsymbol{f})} p(\boldsymbol{f}|\boldsymbol{e})\, p(\boldsymbol{e}) \\ &= \arg\max_{\boldsymbol{e}\in E(\boldsymbol{f})} \sum_{\boldsymbol{d}\in D(G,\boldsymbol{f},\boldsymbol{e})} p(\boldsymbol{d}|\boldsymbol{e})\, p(\boldsymbol{e})\,, \end{aligned} \tag{2.8}$$

where $D(G, \boldsymbol{f}, \boldsymbol{e})$ is a set of derivations of the synchronous grammar G whose source and target language strings are $\boldsymbol{f}$ and $\boldsymbol{e}$, respectively. If the source language syntax on $\boldsymbol{f}$ is available in forms of a parse tree or forest, the derivations are constrained by the tree or forest. Most of the current state-of-the-art SMT systems employ the syntax-based approach in the log-linear framework. One of its important advantage against the PBMT is its structural attribute. The syntax-based approach can model gappy phrase pairs and reordering of large syntactic structure naturally using hierarchical tree structures. They are known to work well for some language pairs requiring long distance reordering, such as German-English, Chinese-English, and Japanese-English.

2.2.4 Language Model

The language model used in the SMT, $p(\boldsymbol{e})$, is a model of generating target language sentences. The most common way to realize the language model is a word n-gram language model. The word n-gram language model assumes that a word is generated according to $(n-1)$ th-order Markov process, in other words, the word generation is conditioned by the preceding $n-1$ words as follows:

$$p(\boldsymbol{e}) = \prod_i p(e_i|e_{i-1}, ..., e_{i-n+1}) \tag{2.9}$$

This is very simple but works effectively, so it has been used also in automatic speech recognition for a long time. Since its naive inference by maximum likelihood estimation faces a serious zero-frequency problem, there are various smoothing methods to give a small probability for unobserved words in a given context. Modified Kneser-Ney smoothing (Chen and Goodman, 1998) is a state-of-the-art method that is commonly used in the field of the SMT. Refer to the literature (e.g., chapter 7 in the book (Koehn, 2010)) for further details.

2.3 Decoding

Decoding is an actual translation process from input source language sentences to target language sentences, using the models described in section 2.2. This section reviews common decoding approaches for the PBMT and SBMT.

2.3.1 Decoding in Phrase-Based MT

One major decoding approach for the PBMT is a left-to-right decoding with beam search over multiple stacks (Koehn et al., 2003). It generates translation hypotheses in the left-to-right order, by choosing and translating source language phrases iteratively within the score range and the stack size constraints of its search space. It uses m stacks for an input sentence with m words, corresponding to the number of words that have been used in the current hypotheses (e.g., the fourth stack holds translation hypotheses generated from four source language words). This stack-based decoding is motivated by the score-based beam search, because the translation hypotheses from the same number of source language words are expected to be in the similar score ranges. Low-scored hypotheses under a some threshold or low-rank hypotheses exceeding its stack size are pruned.

The computational time complexity of this stack-based decoding is $O\left(m \times |\boldsymbol{f}|^2\right)$ with the stack size m. This is still far from efficient for long input sentences. Koehn et al. (2005) introduced a reordering limit to constrain the distance between two source language phrases translated into adjacent target language phrases. This reordering limit reduces the time complexity into $O\left(m \times |\boldsymbol{f}|\right)$, because choices of next phrases at each translation step are limited to a fixed range in the source language sentence and therefore a linear dependency on the sentence length is removed. Although there have been many further improvements in the PBMT decoding, they are beyond the scope of this thesis. Refer to the literature (Koehn, 2010).

2.3.2 Decoding in Syntax-Based MT

Decoding in the SBMT can be regarded as a bilingual parsing problem using the synchronous grammars as discussed in section 2.2.3. There are two different formulations

of the syntax-based MT: the SCFG-based formulation of the HPBMT and SAMT, and the STSG-based formulation of the tree-based MT.

The decoding algorithm is basically similar to well-known chart parsing algotihms, but must consider the target language counterparts as the translation results. The computational time complexity of this parsing-based decoding with beam search is $O\left(|\boldsymbol{f}|^3\right)$, which is still large for long sentences. The typical approach to mitigate this complexity problem is to limit the chart span in the bilingual parsing, which can reduce the complexity but also results in limited range of reordering.

2.4 Evaluation

Evaluation of MT plays a very important role in the development of MT systems. Although subjective (human) evaluation is desirable for meaningful evaluation, it is not easy to evaluate MT results accurately, consistently, and rapidly. Automatic evaluation provides rapid and consistent evaluation, which is suitable for sustainable MT development with frequent system evaluations.

The automatic evaluation basically compares MT results with *reference* translations. Since there is not a unique correct translation for a sentence, the use of several reference translations is desirable for reliable evaluation without dependence on specific expressions. Many automatic evaluation metrics have been proposed with different evaluation strategies. Here several common metrics including those used later in this thesis are reviewed.

2.4.1 Word Error Rate (WER)

Word error rate (WER) is a typical evaluation metric between two sequences of symbols, based on a Levenshtein distance (one of the most common edit distance variants). The WER of a translation hypothesis $\boldsymbol{e}$ for a corresponding reference translation $\boldsymbol{r}$ is the Levenshtein distance from $\boldsymbol{r}$ to $\boldsymbol{e}$ averaged by the length of $\boldsymbol{r}$:

$$\text{WER} = \frac{\text{Lev}\left(\boldsymbol{r}, \boldsymbol{e}\right)}{|\boldsymbol{r}|} = \frac{\text{Sub}\left(\boldsymbol{r}, \boldsymbol{e}\right) + \text{Ins}\left(\boldsymbol{r}, \boldsymbol{e}\right) + \text{Del}\left(\boldsymbol{r}, \boldsymbol{e}\right)}{|\boldsymbol{r}|}, \tag{2.10}$$

where Lev means the Levenshtein distance, Sub, Ins, Del represent the numbers of editing steps of substitutions, insertions, and deletions. The Levenshtein distance is the minimum

number of these steps that can be found efficiently by a dynamic programming method.

The WER is a common evaluation metric especially in automatic speech recognition, but its requirement of matching words in order is too rigid for machine translation because of the translation ambiguity. Och et al. (2001) introduced position-independent word error rate (PER) ignoring this word order problem. The PER is based on simple word matches between two sets of words from the translation hypothesis and the reference translation:

$$\text{PER} = \frac{\text{Sub}_{\text{PI}}\left(\boldsymbol{r}, \boldsymbol{e}\right) + \text{Ins}_{\text{PI}}\left(\boldsymbol{r}, \boldsymbol{e}\right) + \text{Del}_{\text{PI}}\left(\boldsymbol{r}, \boldsymbol{e}\right)}{|\boldsymbol{r}|}, \tag{2.11}$$

where the subscripts PI mean position independent edits. The PER only gives accuracy in lexical choice and is not suitable for evaluating translations between languages with different word order.

2.4.2 Translation Edit Rate (TER)

Both the WER and PER is not suitable for evaluation of translations considering word order variants. Snover et al. (2006) proposed the use of *shift* operation for their metric called translation edit rate (TER). The shift operation moves a word sequence in one operation so that the TER gives a small cost on the move of phrasal structures. The TER is an extension of the WER considering the shift edits as follows:

$$\text{TER} = \frac{\text{Shift}\left(\boldsymbol{r}, \boldsymbol{e}\right) + \text{Sub}\left(\boldsymbol{r}, \boldsymbol{e}\right) + \text{Ins}\left(\boldsymbol{r}, \boldsymbol{e}\right) + \text{Del}\left(\boldsymbol{r}, \boldsymbol{e}\right)}{|\boldsymbol{r}|}, \tag{2.12}$$

where Shift means the number of shift edits.

2.4.3 BLEU

BLEU (Papineni et al., 2002) is a de facto standard metric in the field of SMT that focuses on the precision of local contexts by n-gram precisions. BLEU uses a geometric mean of n-gram precisions of the translation hypothesis at the document level as follows:

$$\text{BLEU}{-}n = \min\left(1, \exp\left(1 - \frac{|\boldsymbol{r}|}{|\boldsymbol{e}|}\right)\right) \times \sqrt[n]{\prod_{m=1}^{n} w_m \, p_m}, \tag{2.13}$$

where p_m is the precision of the word m-grams in the translation hypothesis, and w_m is a weight for the m-gram precision whose sum over all m is one. The first term of the right

side is a penalty term called brevity penalty (BP), which penalizes too short translations with only a limited number of correct n-grams. Most of SMT studies have used BLEU-4 with $n = 4$ and uniform weights $w_m = \frac{1}{n}$ as a standard evaluation metric, referred simply as BLEU. BLEU captures phrasal accuracy of translations in contrast to the word-based evaluation by the WER, PER, and TER. It is known to correlate well with subjective evaluation for not a few language pairs. However, it has very poor correlation with subjective evaluation in patent translation between Japanese and English (Goto et al., 2011), because it overlooks errors in word order.

2.4.4 RIBES

RIBES (Isozaki et al., 2010a; Hirao et al., 2014) is another translation evaluation metric that focuses on the word order problem. RIBES evaluates the correctness of the word order by a rank correlation between the two symbol sequences, the translation hypothesis and the corresponding reference translation as follows:

$$\text{RIBES} = \frac{\tau + 1}{2} \times p_1^{\alpha} \times \text{BP}^{\beta}, \tag{2.14}$$

where α and β are hyperparameters for p_1 (1-gram precision) and BP (brevity penalty) that should be tuned using a small number of tuning data (translation results with reference translations and subjective evaluation results), and τ is a rank correlation coefficient called Kendall's tau. To obtain the rank correlation τ between the translation hypothesis and the reference translation, all their words are aligned one-to-one by a simple heuristic[2] to form two symbol sequences with the same length. Then τ is calculated using the number of word pairs in the translation hypothesis appearing in a concordant and discordant order with the reference translation as follows:

$$\tau = \frac{\#\text{concordant pairs} - \#\text{discordant pairs}}{\frac{1}{2}n(n-1)}, \tag{2.15}$$

where n is the length of the symbol sequence, that is, the denominator is equal to the number of all possible word pairs chosen from the symbol sequence. τ ranges $[-1, 1]$ and is used with a normalization into $[0, 1]$ for RIBES as Equation (2.14).

[2]Refer to the original papers (Isozaki et al., 2010a; Hirao et al., 2014) for details.

RIBES shows very high correlation with subjective evaluation in patent translation between Japanese and English (Goto et al., 2011), with $\alpha = 0.25$ and $\beta = 0.1$.

Chapter 3

Word Segmentation of Domain-Specific Words

Word segmentation is a fundamental problem on natural language applications for languages without explicit word boundaries in their orthography, such as Chinese and Japanese. A word segmenter is usually trained using labeled (word-segmented) corpora in general domains such as newspapers. It does not work well in a different domain such as patents in general, due to many domain-dependent terms that are not covered by the general domain corpora. This causes error propagation into following processes such as the SMT.

Although labeled corpora in the target domain are preferable for accurate domain-dependent word segmentation, they are usually not available in most domains because of the corpus development difficulty. One possible approach to the lack of labeled corpora is an unsupervised method that does not require labeled corpora but uses unlabeled (not word-segmented) corpora. The word segmentation on the unlabeled corpora can be predicted by statistical word boundary clues (Kempe, 1999; Ando and Lee, 2003; Feng et al., 2004) or a model-based inference (Goldwater et al., 2006; Mochihashi et al., 2009). These methods gives good segmentation results in spite of the lack of labeled corpora, but their accuracies are not so high as supervised word segmenters.

Another approach is domain adaptation of the general-domain word segmenter using large-scale unlabeled corpora in the target domain, because the unlabeled corpora gives us distributional information of words as proved by the unsupervised methods. This can

be seen as a semi-supervised learning problem for word segmentation with a small-scale labeled corpus in the general domain and large-scale unlabeled corpora in the target domain. Sun and Xu (2011) proposed a semi-supervised Chinese word segmentation method using labeled and unlabeled corpora in the same domain (newspapers). Guo et al. (2012) adapted it to domain adaptation of Chinese word segmentation, from the newspaper domain to the patent domain. Since a very large number of patents filed so far are publicly available, this semi-supervised approach is promising for the improvement of word segmentation in the patent domain.

Previous studies (Sun and Xu, 2011; Guo et al., 2012) used *accessor variety* (AV) (Feng et al., 2004) as a feature of their discriminative word segmenter. The AV is the number of distinct predecessor and successor characters of a certain string in a given (unlabeled) corpus, which implies word boundaries that have large uncertainty in accessor characters. It is expected to work better with larger corpora by the broader coverage of words. However, it is proportional to the corpus size due to its count-based attribute and not consistent with an intuition of the uncertainty. The previous studies use frequency-based uncertainty classes to normalize the AV values, but it is not straightforward to determine an appropriate setting of classes and threshold values.

To address this problem, this work proposes the use of *branching entropy* (BE) (Jin and Tanaka-Ishii, 2006), the entropy of accessor characters of a certain string. Such a entropy-based metric has been used for unsupervised word segmentation with heuristic thresholds (Kempe, 1999). This work uses the BE as a feature of a discriminative word segmenter instead of the AV. One important advantage of the BE against the AV is its probabilistic attribute; the BE represents the uncertainty in a probabilistic sense regardless of the corpus size. This work further enhances the features by pseudo-dictionary (PD) features derived from the large-scale unlabeled corpus, based on continuous kanji and katakana sequences appearing in the corpus. The proposed method worked effectively in word segmentation experiments for Japanese patent sentences, increasing word segmentation F-measures from 96.87% by a baseline method to 98.36% (47.6% error reduction) without any labeled corpora in the patent domain.

3.1 Conventional Methods

This section firstly describes a character-based word segmentation method based on conditional random fields (CRFs) as the baseline, and then reviews the conventional methods of the domain adaptation.

3.1.1 Baseline Word Segmentation based on Conditional Random Fields

There is a different approach called word-based, which identifies words directly from character sequences by deciding sequentially whether a local character sequence is a word or not. This word-based approach is commonly used in popular Japanese morphological analyzers (JUMAN, ChaSen, MeCab) with their dictionaries to identify in-dictionary words. It has an advantage on the consistency in the segmentation of *known* words included in the dictionary or training data, but usually works less effective for *unknown* words than the character-based approach (Sun, 2010; Wang et al., 2014).

This work uses a character-based word segmenter based on CRFs (Peng et al., 2004; Tseng et al., 2005). It solves a character-based sequential labeling problem. In this work four classes `B`, `M`, `E` (beginning/middle/end of a word), and `S` (single-character word)[1] are used, as Sun and Xu (2011).

The baseline features follow the work of Japanese word segmentation by Neubig et al. (2011): label bigrams, character n-grams (n=1, 2), and character type n-grams (n=1, 2, 3), within [i-2, i+2] for classifying the word at the position i. The character types are *kanji*, *katakana*, *hiragana*, digits, roman characters, and others.

3.1.2 Word Segmentation Adaptation using Accessor Variety

Sun and Xu (2011) and Guo et al. (2012) used Accessor Variety (AV) (Feng et al., 2004) derived from unlabeled corpora as word segmentation features. AV is a word extraction criterion from un-segmented corpora, focusing on the number of distinct characters appearing

[1] Guo et al. (2012) used six classes including `B2`, `B3` (second and third character in a word) proposed by Zhao et al. (2006) for Chinese word segmentation. This work uses the four classes, because the six classes did not improve the word segmentation accuracy in my pilot test.

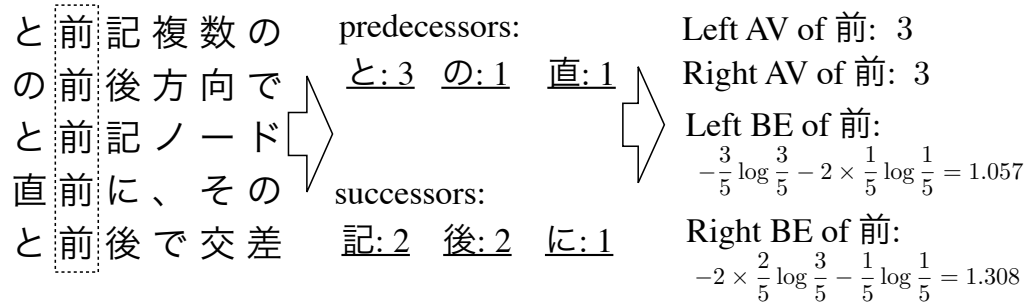


Figure 3.1: Example of accessor variety (AV) and branching entropy (BE) for a character "前".

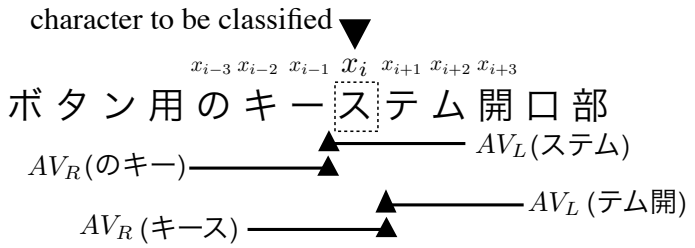


Figure 3.2: Accessor variety features on the character x_i.

around a string. The AV of a string $\boldsymbol{x}_n$ is defined as

$$AV(\boldsymbol{x}_n) = \min\left\{AV_L(\boldsymbol{x}_n), AV_R(\boldsymbol{x}_n)\right\}, \tag{3.1}$$

where $AV_L(\boldsymbol{x}_n)$ is the left AV (the number of distinct predecessor characters) and $AV_R(\boldsymbol{x}_n)$ is the right AV (the number of distinct successor characters). The AV-based word extraction is based on an intuitive assumption; *a word appears in many different context so that there is a large variation of its accessor characters*. Intuitively this assumption seems true. Figure 3.1 shows an example of the AV calculation for a character "前". If the character is a word by itself, it is expected to appear in many different context so that the AV values become large by different accessor characters. Note that the AV values are frequency-based and proportional to the corpus size in general. Previous studies use several frequency classes with corresponding threshold values tuned according to the corpus, but it is not straightforward to determine appropriate classes and threshold values.

Sun and Xu (2011) used the following features based on the left and right AVs of character n-grams for classifying x_i, which imply word boundaries around x_i, as illustrated in

Figure 3.2:

- Left AV of n-gram starting from x_i: $AV_L(x_i, ..., x_{i+n-1})$,
- Right AV of n-gram ending with x_{i-1}: $AV_R(x_{i-n}, ..., x_{i-1})$,
- Left AV of n-gram starting from x_{i+1}: $AV_L(x_{i+1}, ..., x_{i+n})$, and
- Right AV of n-gram ending with x_i: $AV_R(x_{i-n+1}, ..., x_i)$,

in addition to their baseline features such as character n-grams.

The AV values are calculated on the labeled and unlabeled corpora and classified into several frequency classes for AV features. Sun and Xu (2011) used absolute frequency thresholds, "$>$ 50" (if the AV exceeds 50), "30 - 50" (if the AV is between 30 and 50), and the actual values (if the AV is not greater than 30), and used them as binary bucket features. Guo et al. (2012) used different relative frequency classes: `H`, `M`, `L` for top 5%, between top 5% and 20%, below top 20%. This kind of frequency-based grouping quantizes the AV values. The thresholds and the number of the classes are tuned using some held-out data (Sun and Xu, 2011) or chosen empirically (Guo et al., 2012). Such a tuning is not easy in general, especially with a large number of the classes. The relative frequency classes of Guo et al. (2012) are used in the following experiments.

These AV features give word boundary clues to the CRF-based word segmenter. In its training, the AV features are associated with classes in the labeled data. Intuitively, a high left AV value suggests word boundary at the left of the target character and is associated with `B` and `S`, and a low left AV value is associated with `M` and `E` in contrast. In the test phase, the AV features help predict classes even in contexts that are not found in the labeled data, while the baseline surface-based features are not effective in such *unseen* contexts. The AV-based word boundary clues are expected to be reliable for many different domain-specific words when large-scale unlabeled corpora in the target domain are available.

3.2 Proposed Word Segmentation Adaptation Method

This work proposes a word segmentation adaptation method using two additional novel types of features: branching entropy (BE) features and pseudo-dictionary (PD) features.

Its semi-supervised learning framework is the same as Sun and Xu (2011) and Guo et al. (2012). The BE features are practically useful because of the probabilistic attribute of the BE, and the PD features reflect characteristics of Japanese compound words.

3.2.1 Branching Entropy Features

The BE (Jin and Tanaka-Ishii, 2006) is a different word boundary clue based on probabilistic uncertainty of accessor characters. Jin and Tanaka-Ishii (2006) used the BE for unsupervised Chinese word segmentation. Their approach is based on an intuitive assumption; *the uncertainty of successive characters is large at a word boundary*. The uncertainty of the successive character X after a given string $\boldsymbol{x}_n = x_1...x_n$ of the length n can be measured by the BE as the local conditional entropy of X with $\boldsymbol{X}_n$ instantiated:

$$H(X|\boldsymbol{X}_n = \boldsymbol{x}_n) = -\sum_{x \in V_x} P(x|\boldsymbol{x}_n) \log P(x|\boldsymbol{x}_n), \tag{3.2}$$

where $\boldsymbol{X}_n$ is the context of the length n, and V_x is a set of characters. Jin and Tanaka-Ishii (2006) used the BE around character n-grams: left BE $H_L(\boldsymbol{x}_n)$ for predecessor characters and right BE $H_R(\boldsymbol{x}_n)$ for successor characters. Figure 3.1 also shows an example of the BE calculation. The left and right BE values are slightly different due to the different distributions of predecessor and successor characters. Even if the number of distinct accessor characters is large, the probabilistic certainty varies with their variance and is not necessarily large. Another important advantage of the BE is its probabilistic attribute. The uncertainty of accessor characters represented by a certain BE value is basically the same even for different corpus sizes, while the AV values increase with the corpus size in general.

The BE features are binary bucket features based on rounded integer values of the left and right BEs of character n-grams, similarly defined as the AV features illustrated in Figure 3.2. This simple quantization is motivated by the probabilistic attribute of the BE.

- Left BE of n-gram starting from x_i: $H_L(x_i, ..., x_{i+n-1})$
- Right BE of n-gram ending with x_{i-1}: $H_R(x_{i-n}, ..., x_{i-1})$
- Left BE of n-gram starting from x_{i+1}: $H_L(x_{i+1}, ..., x_{i+n})$
- Right BE of n-gram ending with x_i: $H_R(x_{i-n+1}, ..., x_i)$

3.2.2 Pseudo-dictionary Features

This work also uses Japanese-oriented heuristic word boundary clues, based on characteristics of Japanese compound words. Compound words in Japanese patents are usually written in kanji (for Japanese- or Chinese-origin words) or katakana (for imported words from Western languages). Most of their component words are also used individually and in different compound words. For example, a katakana word "ステム" (stem) is used in many compound words such as "キーステム" (key stem) and "ステムセル" (stem cell). Appearance of a distinct katakana sequence "ステム" implies word boundaries between "キー" and "ステム" and between "ステム" and "セル". Such a word boundary clue may help to identify component words appearing in different contexts. The motivation of these intuitive word boundary clues based on the character type is similar to the use of punctuations as reliable word boundaries by Sun and Xu (2011).

To include such information, distinct kanji and katakana sequences are used as pseudo-dictionary entries. The definition of the pseudo-dictionary features follows the dictionary word features used in Japanese morphological analyzer KyTea[2]: whether or not the character is in the beginning/middle/end of one of the dictionary words of a certain length. An example of the pseudo-dictionary features is shown in Figure 3.3. The character "ス" in the example has a feature "L3_katakana", representing the character is located at the leftmost position of a matched katakana pseudo-dictionary word of the length of three characters "ステム". Two distinguished pseudo-dictionaries for kanji and katakana are used for the PD features. Short sequences whose length is shorter than 2 characters for kanji and 3 characters for katakana are excluded from the pseudo-dictionaries. The length of the long pseudo-dictionary words exceeding five characters is labeled as "5+" to mitigate feature sparseness.

3.3 Experiments

Experiments were conducted using the NTCIR PatentMT data to evaluate the word segmentation accuracy by the proposed word segmentation adaptation and compared it with

[2]http://www.phontron.com/kytea/method.html

Figure 3.3: Example of pseudo-dictionary features.

Table 3.1: Corpus statistics for word segmentation experiments.

Dataset	Type	#sentences	#Ja characters	#words
BCCWJ	Labeled (Training)	53,899	1,810,675	1,242,137
BCCWJ$_{UL}$	Unlabeled	6,017,627	185,289,168	n/a
NTCIR	Unlabeled	3,191,228	214,963,715	n/a
PatentJP	Unlabeled	537,494,485	42,175,165,488	n/a
Test$_{Patent}$	Labeled (Test)	2,000	127,825	81,481
Test$_{BCCWJ}$	Labeled (Test)	6,406	201,080	135,664

those by other methods.

3.3.1 Setup

The word segmenters were implemented using the features described in the previous section, with CRFsuite[3] and its default hyperparameters. The CORE data of Balanced Corpus of Comtemporary Written Japanese (Maekawa, 2007) were used as the labeled general domain corpus for training the word segmentation model[4], split by about 9:1 for training (BCCWJ) and test (Test$_{BCCWJ}$) sets. For unlabeled corpus, its non-CORE portion (BCCWJ$_{UL}$), the Japanese portion of NTCIR-9 PatentMT (Goto et al., 2011) Japanese-English bitext (NTCIR), and Japanese monolingual patent corpus provided for NTCIR-9 PatentMT (PatentJP), are used. The test set in the patent domain (Test$_{Patent}$) was in-house 2,000 sentences in which the word segmentation was manually annotated by the same word segmentation standards as the other labeled data. Corpus statistics are shown in Table 3.1.

[3] http://www.chokkan.org/software/crfsuite/

[4] Kanji numbers were replaced with digits for consistency with the patent corpus.

3.3.2 Compared Methods

The following word segmentation features were compared in the word segmentation experiments.

- Baseline: only the baseline features described in 4.1
- +AV: the AV (n=2,3,4,5) and baseline features
- +BE: the BE (n=1,2,3,4,5) and baseline features
- +PD: the PD and baseline features
- +BE +PD: the BE (n=1,2,3,4,5), PD, and baseline features

To investigate the impact of the unlabeled corpus size in the semi-supervised approach, Two different conditions, mid-scale and large-scale, were compared; BCCWJ$_{UL}$ and NTCIR were used in the mid-scale condition, and BCCWJ$_{UL}$ and PatentJP[5] in the large-scale condition. Here, the pseudo-dictionaries of kanji and katakana sequences were composed by kanji and katakana sequences found in the unlabeled data. The word segmenters with a publicly available word segmenter were also compared with the CRF-based segmenters for reference: MeCab[6] with a model based on a Japanese dictionary UniDic[7], and KyTea[8] with its attached model.

3.3.3 Results

Table 3.2 shows word segmentation results in F-measures in the patent and general (BCCWJ) domains, and recalls of out-of-vocabulary words (OOV recall) in the patent domain focusing on domain-specific words not included in the general domain corpus. All the additional features showed better results in the patent domain than the baseline features and MeCab, which were statistically significant (p=0.05) by bootstrap resampling tests.

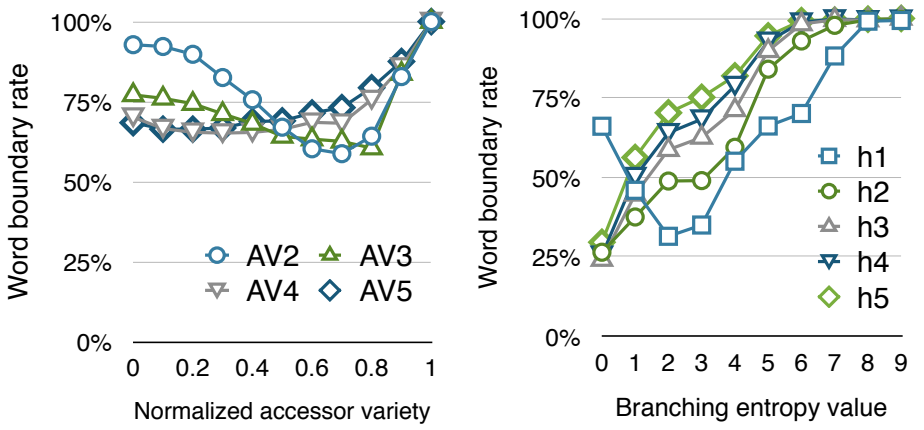
The AV and BE features helped to outperform MeCab in the patent domain especially in the OOV recall while the baseline performance was much worse. The BE features worked

[5]The patent sentences in NTCIR is also included in PatentJP.
[6]https://code.google.com/p/mecab/
[7]http://sourceforge.jp/projects/unidic/
[8]http://www.phontron.com/kytea/

Table 3.2: Word segmentation F-measures (%) for the patent and original domains and OOV recalls (%) in the patent domain. ^{A}, ^{B}, and ^{P} indicate significantly better results than +AV, +BE, +PD (in the same group), ^{M} and ^{L} indicate significantly better results than mid- and large-scale. PD_m means the PD features derived from the mid-scale unlabeled corpora.

Condition	Feature	Patent		BCCWJ
		F-measure (%)	OOV Recall (%)	F-measure (%)
Labeled	Baseline	96.87	87.94	97.85
Unlabeled (Mid-scale)	+AV	^{L}98.08	91.25	98.27
	+BE	^{A}98.25	91.58	98.38
	+PD	^{L}97.85	91.18	98.08
	+BE +PD	^{A,B}98.32	92.09	98.39
Unlabeled (Large-scale)	+AV	^{P}97.80	90.79	98.26
	+BE	^{A,P,M}98.34	91.62	98.33
	+PD	97.12	89.32	98.36
	+BE +PD	^{A,P}98.32	92.33	98.37
	+BE +PD_m	^{A,P,M}**98.36**	**92.61**	98.37
MeCab		97.73	86.94	98.35
KyTea		96.32	83.99	97.94

consistently with the different corpus sizes. The AV features with the large-scale data showed obviously worse results than with the mid-scale data; this indicates instability of the AV features with different corpus sizes. The PD features showed good performance especially in OOV recall, but those from the large-scale corpora did not work so well. This is possibly due to inappropriate pseudo-dictionary entries extracted around typographical errors, which sometimes occur between characters with similar type faces. Thus I additionally tested the combination of the PD features in the mid-scale condition and the large-scale BE features, and that showed the best results. This indicates my domain adaptation is very effective for domain-specific words.



(a) Accessor variety (normalized with the maximum value)

(b) Branching entropy

Figure 3.4: Word boundary rate by quantized accessor variety and branching entropy values in patent domain.

reference: |前_記|待_機|ト_レ_イ|
(afore-mentioned) (standby) (tray)
baseline: |前|記_待|機|ト_レ_イ|
Δentropy: |前_記|待_機|ト_レ_イ|
▲ △ ▲ △ ▲ △ △ ▲

(a) A baseline error is corrected by the proposed feature

reference: |逆|相|の|パ_ル_ス|
(opposite) (of) (pulse)
(phase)
baseline: |逆_相|の|パ_ル_ス|
Δentropy: |逆_相|の|パ_ル_ス|
▲ △ ▲ △ △ ▲

(b) A baseline error is not corrected by the proposed feature

reference: |フ_ィ_ー_ド_バ_ッ_ク|ル_ー_プ|
(feedback) (loop)
baseline: |フ_ィ_ー_ド_バ_ッ_ク|ル_ー_プ|
Δentropy: |フ_ィ_ー_ド|バ_ッ_ク|ル_ー_プ|
▲ △ △ △ ▲ △ △ ▲ △ △ ▲

(c) The proposed feature derives an error

Figure 3.5: Word segmentation examples. Vertical bars ("|") represent word boundaries, underbars ("_") represents non-boundaries. Black and white triangles indicates character boundaries with positive and negative branching entropy differences, respectively.

3.3.4 Detailed Analysis

For detailed analysis of the difference between the AV and BE, their correlation with word boundaries was investigated. If the BE had good correlation, it supports the results above. The correlation between their values and word boundary rate were analyzed for each entropy value. The word boundary rate was defined as follows using the number of character and word boundaries whose quantized BE or AV value is m:

$$\text{Word boundary rate}(m) = \frac{\#\text{word boundaries}(m)}{\#\text{character boundaries}(m)}. \tag{3.3}$$

Figure 3.4 shows the word boundary rate for corresponding AV and BE values in the large-scale condition. The AV values were normalized with the maximum value of each n-gram AV. The x-axis labels represent the quantized values.

Different characteristics by the accessor variety values and the entropy values can be observed from the figures. The AV in Figure 3.4(a) showed poor correlation with word boundaries in general. The AV of higher n-gram order seem to correlate with the word boundary rate to some extent, but they are not enough to determine word boundaries. The BE itself shown in Figure 3.4(b) seemed to correlate with word boundaries. It worked differently according to its context length; higher branching entropy values were needed to predict word boundaries with a shorter context. This implies some relativity of the branching entropy. The branching entropy with a shorter context would be relatively high in average, compared to that with a longer context.

3.3.5 Segmentation Examples

Figure 3.5 shows word segmentation examples. by the baseline and patent-adapted segmenters with the branching entropy difference features. Figure 3.5(a) shows a typical error correction example, in which wrong segmentation of a *kanji* compound word was corrected. Many compound words appear in patent documents as technical terms and are often segmented incorrectly, so this kind of error correction helps to improve the word segmentation performance. Others are negative examples. In Figure 3.5(b), the compound word 逆相 (opposite phase) was not segmented. This kind of under-segmentation errors may occur when the component words do not appear frequently in different contexts. Figure 3.5(c) shows an

over-segmentation error on compound words. The word フィードバック (feedback) was recognized incorrectly as a compound word of フィード (feed) and バック (back). The number of this kind of errors were not so large in our experiments, but they are important problems for further studies.

3.4 Related Work

There are two major approaches for domain adaptation of word segmentation: active learning using additional labeled data, and semi-supervised learning using unlabeled data as this work.

The active learning approach learns correct word segmentations from a small number of additional human annotations. Tsuboi et al. (2008) presented an extended training algorithm for Conditional Random Fields using partially labeled data. Neubig et al. (2011) proposed an efficient pointwise prediction approach suitable for iterative active learning. This approach is effective but requires additional human efforts.

On the other hand, the semi-supervised approach utilizes unlabeled data. Wang et al. (2011) used an existing segmenter to extract n-gram frequency features on large-scale unlabeled data. Sun and Xu (2011) compared substring-wise mutual information and the AV. These studies did not focus on the problem of domain adaptation, but their techniques can be directly applied to domain adaptation. Guo et al. (2012) used the AV for domain adaptation of Chinese word segmentation on patent documents by a method similar to (Sun and Xu, 2011). The semi-supervised approach has an advantage of automatic adaptation without human efforts.

The semi-supervised approach also relates to unsupervised methods with respect to word boundary clues. Ando and Lee (2003) proposed mostly unsupervised Japanese compound word segmentation using character n-gram statistics. The AV and BE were originally proposed for unsupervised word segmentation (Feng et al., 2004; Jin and Tanaka-Ishii, 2006). Zhikov et al. (2010) extended the BE-based method using Minimum Description Length. These studies used different clues of word and its boundary, but they are not sufficiently accurate (about 80% in F-measure). Mochihashi et al. (2009) proposed Bayesian word segmentation, which is fully model-based method — not based on intuitive word seg-

mentation clues. Word segmentation obtained by such unsupervised methods may not be consistent with human annotation standards used in existing natural language processing components, so the semi-supervised approach is expected to be suitable for practical use.

3.5 Conclusion

This chapter presented an effective domain adaptation technique for discriminative word segmentation in the patent domain. The BE features worked much better and were more stable than the AV features in the experiments, thanks to the probabilistic attribute of the BE. The word segmenter further improved by the PD features especially in terms of the OOV recall, which is important for practically useful word segmentation in the patent domain; its performance in word segmentation F-measure was fairly high exceeding 98%, even without labeled corpus in the patent domain.

Chapter 4

Transliteration of Technical Terms

Transliteration is used for providing translations for source language words that have no appropriate counterparts in a target language, such as certain technical terms and named entities. Statistical machine transliteration (Knight and Graehl, 1998) is a technology designed to solve this problem in a statistical manner. Bilingual dictionaries can be used to train its model, but many of their entries are actually *translations* but not *transliterations*. Such non-transliteration pairs hurt the transliteration model and should be eliminated in advance.

Sajjad et al. (2012) proposed a method for identifying such non-transliteration pairs, and applied it successfully to *noisy* word pairs obtained from automatic word alignment using bilingual corpora. It enables a statistical machine transliteration to be bootstrapped from bilingual corpora. This approach is beneficial because it does not require carefully developed bilingual transliteration dictionaries and it can learn domain-specific transliteration patterns from bilingual corpora in the target domain. However, their transliteration mining approach is sample-wise; that is, it decides whether or not a bilingual phrase pair is transliteration. There can be cases where a part of the phrase is the transliteration of the other. For example, suppose a Japanese transliterated compound word キーステム (key stem) is aligned only to an English word "stem". This word pair consists a transliteration fragment ⟨ステム(stem), stem⟩ and a non-transliteration part キー (key) in the Japanese side as partial noise. The sample-wise method cannot extract the transliteration fragment, but it can only accept or reject the whole word pair. Such a sample-wise decision is difficult

due to the trade-off between the transliteration fragment and partial noise, and often accepts partially noisy pairs incorrectly, which introduce noise into the training data for statistical machine transliteration.

This work proposes a novel method for extracting such transliteration fragments. The method uses a noise-aware character alignment model that distinguishes non-transliteration (noise) parts from transliteration (signal) parts. The model is an extension of a Bayesian alignment model (Finch and Sumita, 2010) and can be trained by using a sampling algorithm extended for a constraint on noise. In experiments involving Japanese-to-English transliteration bootstrapping using patent data, the proposed method showed much better partial noise identification performance than an IBM-model-based baseline using NULL alignments, and achieved better transliteration accuracy than the sample-wise transliteration mining method (Sajjad et al., 2012).

4.1 Bayesian Many-to-many Alignment

First the Bayesian many-to-many character alignment technique proposed by (Finch and Sumita, 2010) is reviewed briefly.

4.1.1 Model

Their model is a bilingual extension of the unigram Dirichlet Process (DP) for unsupervised word segmentation (Goldwater et al., 2006; Xu et al., 2008), based on a generative process of bilingual string pairs. The probability of a bilingual string pair $\langle \boldsymbol{s}, \boldsymbol{t} \rangle = \langle s_1 ... s_{|\boldsymbol{s}|}, t_1 ... t_{|\boldsymbol{t}|} \rangle$ is the sum of the probabilities of its all possible co-segmentations:

$$p(\langle \boldsymbol{s}, \boldsymbol{t} \rangle) = \sum_{\gamma \in \Gamma(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)} p(\langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle, ..., \langle \bar{\boldsymbol{s}}_{K_\gamma}, \bar{\boldsymbol{t}}_{K_\gamma} \rangle) = \sum_{\gamma \in \Gamma(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)} \prod_{1 \leq k \leq K_\gamma} p(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle) \tag{4.1}$$

where γ is a co-segmentation over $\langle \boldsymbol{s}, \boldsymbol{t} \rangle$, a sequence of K_γ bilingual substring pairs $\langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle, ..., \langle \bar{\boldsymbol{s}}_{K_\gamma}, \bar{\boldsymbol{t}}_{K_\gamma} \rangle$, and $\Gamma(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)$ is the set of all possible co-segmentations. $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$ $(1 \leq k \leq K_\gamma)$ is a substring pair in the co-segmentation that can be regarded as many-to-many aligned characters. These substring pairs are considered to be generated from the DP in this model. The DP

for $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$ in a bilingual string pair can be denoted as follows:

$$\begin{aligned} G|_{\alpha, G_0} &\sim \mathrm{DP}(\alpha, G_0) \\ \langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle | G &\sim G, \end{aligned} \tag{4.2}$$

where G is a probability distribution over substring pairs according to a DP prior with base measure G_0 and hyperparameter α. G_0 is modeled as a joint spelling model relying only on the lengths of $\bar{\boldsymbol{s}}_k$ and $\bar{\boldsymbol{t}}_k$ (denoted as $|\bar{\boldsymbol{s}}_k|$ and $|\bar{\boldsymbol{t}}_k|$) as follows:

$$G_0\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle\right) = \frac{\lambda_s^{|\bar{\boldsymbol{s}}_k|}}{|\bar{\boldsymbol{s}}_k|!} e^{-\lambda_s} v_s^{-|\bar{\boldsymbol{s}}_k|} \times \frac{\lambda_t^{|\bar{\boldsymbol{t}}_k|}}{|\bar{\boldsymbol{t}}_k|!} e^{-\lambda_t} v_t^{-|\bar{\boldsymbol{t}}_k|}. \tag{4.3}$$

This is a simple joint probability of two spelling models. In each spelling model, each alphabet appears based on a uniform distribution over the vocabulary (of size v_s or v_t), and each substring length follows a Poisson distribution (with the average length λ_s or λ_t) (Brown et al., 1992). The model handles an infinite number of substring pairs according to the Chinese Restaurant Process (CRP). The probability of a substring pair $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$ drawn from the DP is based on the counts of all other substring pairs as follows:

$$p\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle \mid \{\langle \bar{\boldsymbol{s}}_i, \bar{\boldsymbol{t}}_i \rangle\}_{-k}\right) = \frac{N\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle\right) + \alpha G_0\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle\right)}{\sum_i N\left(\langle \bar{\boldsymbol{s}}_i, \bar{\boldsymbol{t}}_i \rangle\right) + \alpha}. \tag{4.4}$$

Here $\{\langle \bar{\boldsymbol{s}}_i, \bar{\boldsymbol{t}}_i \rangle\}_{-k}$ means a set of substring pairs observed so far (not including $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$), and $N\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle\right)$ is the number of appearance of $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$ in the substring pair set. This model is suitable for representing a very sparse distribution over arbitrary substring pairs, thanks to reasonable CRP-based smoothing for unseen pairs based on the spelling model.

Note that two different kinds of probabilities are maintained: the DP-based probability of a substring pair $p\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle \mid \{\langle \bar{\boldsymbol{s}}_i, \bar{\boldsymbol{t}}_i \rangle\}_{-k}\right)$ in Equation (4.4), and the marginal probability of a string pair $p(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)$ in Equation (4.1) considering all possible co-segmentations.

4.1.2 Sampling-based Inference

The goal of this method is to find the best co-segmentation for each bilingual string pair in training data $D = \{\langle \boldsymbol{s}, \boldsymbol{t} \rangle_m \mid 1 \leq m \leq M\}$:

$$\hat{\boldsymbol{\gamma}} = \arg\max_{\check{\boldsymbol{\gamma}}} p\left(\check{\boldsymbol{\gamma}} \mid D\right) = \arg\max_{\check{\boldsymbol{\gamma}}} \prod_{m=1}^{M} p\left(\gamma_m \mid D\right) = \arg\max_{\boldsymbol{\gamma}} \prod_{m=1}^{M} \prod_{k=1}^{K_{\gamma m}} p\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle_m \mid D\right). \tag{4.5}$$

where $\tilde{\boldsymbol{\gamma}} = \{\gamma_m = \langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle_m, ..., \langle \bar{\boldsymbol{s}}_{K_{\gamma_m}}, \bar{\boldsymbol{t}}_{K_{\gamma_m}} \rangle_m \mid 1 \leq m \leq M\}$[1] is a set of co-segmentations for D. To approximate the true posterior distribution for $p(\gamma_m \mid D)$, (Finch and Sumita, 2010) used an efficient forward-backward inference with a blocked Gibbs sampling algorithm, called forward filtering backward sampling (FFBS) (Scott, 2002; Mochihashi et al., 2009). It enables efficient block-wise sampling over true posterior distributions, by employing an efficient dynamic programming-based calculation similar to the well-known forward-backward algorithm. The blocked Gibbs sampler samples a co-segmentation $\gamma_m = \langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle_m, ..., \langle \bar{\boldsymbol{s}}_{K_{\gamma_m}}, \bar{\boldsymbol{t}}_{K_{\gamma_m}} \rangle_m$ at the same time, using the posterior distributions conditioned by the sample space of co-segmentations on the other training data $S_{-m} = \{\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle_{m'} \mid 1 \leq m' \leq M,\ m' \neq m,\ 1 \leq k \leq K_{\gamma_{m'}}\}$, where $K_{\gamma_{m'}}$ is the number of substring pairs in the co-segmentation on m'-th data. I denote the DP-based probability of a substring pair in Equations (4.4) and (4.5) conditioned by S_{-m} as:

$$p_{-m}(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle) \equiv p(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle_m \mid S_{-m}). \tag{4.6}$$

Algorithm 1 shows the algorithm for the blocked Gibbs sampling. It starts the training from random co-segmentations over the training data, and samples and updates the co-segmentation of each bilingual string pair iteratively by the FFBS. Final co-segmentation results are obtained as a set of sampled substring pair sequences that maximize the probability in Equation (4.5) after convergence. Here the FFBS for a bilingual string pair $\langle \boldsymbol{s}, \boldsymbol{t} \rangle_m$ is explained, shown in the innermost loop in Algorithm 1 (lines 4 to 7). Hereafter, the index m is omitted for readability except for p_{-m}.

Forward filtering

In the forward filtering step, forward probabilities are calculated (line 6). The forward probability $\alpha(I, J)$ is the probability of a bilingual substring pair with the length of I and J ($\langle s_1 ... s_I, t_1 ... t_J \rangle$), defined in a recursive manner considering all possible substring pairs $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$ ending with the indices (I, J):

$$\alpha(I, J) = \sum_{\bar{\boldsymbol{s}}_k \in C(I, \boldsymbol{s}),\ \bar{\boldsymbol{t}}_k \in C(J, \boldsymbol{t})} p_{-m}(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle) \times \alpha(I - |\bar{\boldsymbol{s}}_k|, J - |\bar{\boldsymbol{t}}_k|), \tag{4.7}$$

[1] Here $\tilde{\boldsymbol{\gamma}}$ is used for a set of co-segmentations to distinguish it with a co-segmentation γ.

Algorithm 1 The blocked Gibbs sampling for Bayesian unsupervised alignment.

Require: Bilingual string pairs $D = \{\langle \boldsymbol{s}, \boldsymbol{t} \rangle_m \mid 1 \leq m \leq M\}$

Ensure: Many-to-many aligned bilingual string pairs $= \{\langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle_m, ... \langle \bar{\boldsymbol{s}}_{K_m}, \bar{\boldsymbol{t}}_{K_m} \rangle_m \mid 1 \leq m \leq M\}$

1: Initialize $\langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle_m, ... \langle \bar{\boldsymbol{s}}_{K_m}, \bar{\boldsymbol{t}}_{K_m} \rangle_m$ randomly for all m

2: **repeat**

3: **for** m in $1, ..., M$ (in random order) **do**

4: Remove current co-segmentation $\langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle_m, ... \langle \bar{\boldsymbol{s}}_{K_m}, \bar{\boldsymbol{t}}_{K_m} \rangle_m$ from the sample space

5: Initialize forward probability matrix $A = [\alpha(i, j)]$ $(0 \leq i \leq |\boldsymbol{s}|,\ 0 \leq j \leq |\boldsymbol{t}|)$ with $A_{0,0} \leftarrow 1$ and `undefined` for others

6: Compute and store the forward probabilities in A recursively by the dynamic programming {forward filtering}

7: Sample bilingual substring pairs from the end as $\langle \bar{\boldsymbol{s}}_{K_m}, \bar{\boldsymbol{t}}_{K_m} \rangle_m, ... \langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle_m$, and update the co-segmentation {backward sampling}

8: **end for**

9: **until** the number of iterations reaches its limit

where $\alpha(0, 0) = 1$ (line 5) and $C(I, \boldsymbol{s})$ is a set of all possible substrings from a string $\boldsymbol{s}$ that end with s_I[2]. Note that the DP-based probability in Equation (4.4) is conditioned by S_{-m} here (line 4). This process calculates the forward probability at the end of the bilingual string pair $\alpha(|\boldsymbol{s}|, |\boldsymbol{t}|)$, that is also equivalent to the probability of the whole string pair $p(\langle \boldsymbol{s}, \boldsymbol{t} \rangle)$. The forward probabilities with all the other indices are calculated by its recursive definition. Since the forward probability at a certain position is calculated only once by a dynamic programming using a matrix A, these forward probabilities can be obtained efficiently. Figure 4.1 (a) shows an example of the forward filtering, in which each arrow represents the corresponding substring pair. At first the calculation of the forward probability of the whole bilingual string pair $\langle$カバー, $cover\rangle$ is called. It calls the calculation of the forward probabilities at preceding positions recursively as Equation (4.7). The probabilities along with the paths are accumulated when they gather into the same positions, and

[2]This work considers arbitrary-length substrings up to s_I, $\{s_1 ... s_I,\ s_2 ... s_I,\ ...,\ s_{I-1} s_I,\ s_I\}$, following (Finch and Sumita, 2010).

finally $p\left(\langle カバー, cover\rangle\right)$ is obtained considering its all possible co-segmentations.

Backward sampling

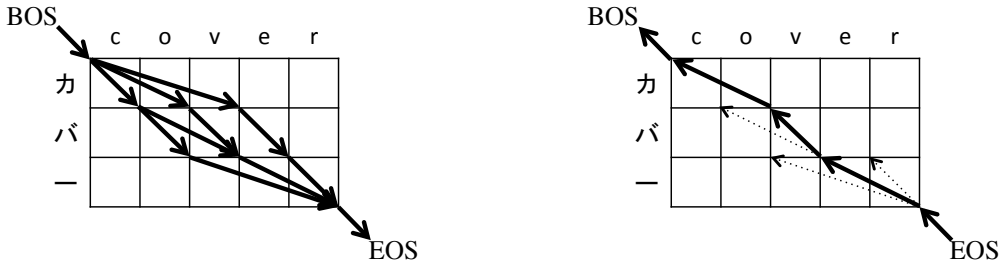
Next in the backward sampling step, the co-segmentation of the bilingual string pair is determined from its end, by the sampling based on the posterior distribution of bilingual substring pairs (line 7). The posterior probability of the bilingual substring pair at the end of the whole string pair, $\langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle$, can be calculated using its DP-based probability and its forward probability as follows:

$$p\left(\langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) = \frac{p_{-m}\left(\langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle\right) \times \alpha\left(|\boldsymbol{s}| - |\bar{\boldsymbol{s}}_K|, |\boldsymbol{t}| - |\bar{\boldsymbol{t}}_K|\right)}{p\left(\langle \boldsymbol{s}, \boldsymbol{t} \rangle\right)}. \tag{4.8}$$

Here, the backward probability is constant (1.0) because of the unigram-based formulation of this model. The numerator of the right side of the equation is the sum of the probabilities of all possible co-segmentations that include $\langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle$ at the end, and its denominator is the probability of the whole string pair, which can be considered as a constant in comparison of different $\langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle$ at the end. Thus, a substring pair $\langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle$ can be sampled based on the true posterior distribution as follows:

$$p\left(\langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle\right) \times \alpha\left(|\boldsymbol{s}| - |\bar{\boldsymbol{s}}_K|, |\boldsymbol{t}| - |\bar{\boldsymbol{t}}_K|\right). \tag{4.9}$$

Once the substring pair at the end is determined, its preceding substring pair $\langle \bar{\boldsymbol{s}}_{K-1}, \bar{\boldsymbol{t}}_{K-1} \rangle$ can be sampled in the same manner, regarding its backward probability as a constant. The sampling can be repeated by the beginning of the bilingual string pair, and finally its co-segmentation $\langle \bar{\boldsymbol{s}}_1, \bar{\boldsymbol{t}}_1 \rangle, ... \langle \bar{\boldsymbol{s}}_K, \bar{\boldsymbol{t}}_K \rangle$ is obtained. Figure 4.1 (b) shows an example of the backward sampling. Substring pairs are sampled among all possible substring pairs ending with the current position (starting from the end of the bilingual string pair), based on their posterior probabilities in Equation (4.9). The sampling is repeated by the beginning as shown in the figure (bold solid arrows represent sampled substring pairs and dotted arrows are those not sampled), and a co-segmentation $\langle カ, co \rangle, \langle バ, v \rangle, \langle ー, er \rangle$ is obtained.



(a) Forward filtering (b) Backward sampling

Figure 4.1: Forward filtering backward sampling in Finch et al. (2010). (A few example paths are shown.)

4.2 Proposed Method

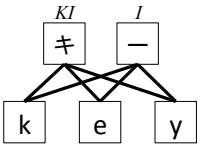
This work proposes an extended many-to-many alignment model that can handle partial noise. The model described in the previous section is extended by introducing a noise symbol and state-based probability calculation.

4.2.1 Partial Noise in Transliteration Data

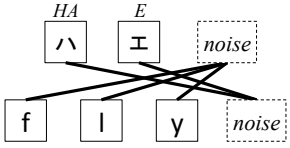
Figure 4.2 shows transliteration examples with "no noise," "sample-wise noise," and "partial noise." The solid lines in the figure show correct many-to-many alignment links. Examples (a) and (b) are distinguished effectively by (Sajjad et al., 2012). The proposed method aims to realize alignment as in examples (c) and (d) by distinguishing its non-transliteration (noise) part, which cannot be achieved with the existing methods. Here, it additionally use *NULL* symbols in Figure 4.2(e) that are expected to be aligned to white spaces and silent letters (such as "b" in "doubt") in signal parts. Characters aligned to the NULL symbols are needed to learn phrasal transliteration (at the character level), while characters aligned to the noise symbols can be eliminated. This work defines noise and NULL as follows[3]:

Noise Substrings whose pronunciations do not appear in the transliterated strings, and

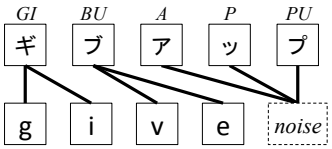
[3]There are often common but non-equivalent transliteration examples such as "McDonald's" and its transliteration to Japanese, マクドナルド (MA KU DO NA RU DO). "'s" is regarded as partial noise in this work and the correct (back-)transliteration of マクドナルド as "McDonald."



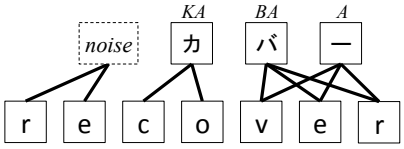
(a) no noise



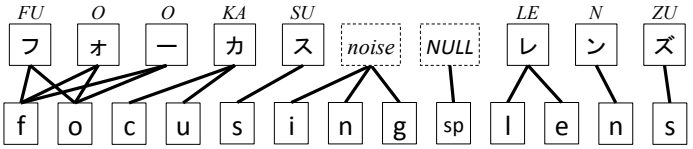
(b) sample-wise noise



(c) partial noise: English should be "give up"



(d) partial noise: Japanese should be "リカバー"



(e) partial noise: Japanese should be "フォーカシングレンズ"

Figure 4.2: Three types of noise in transliteration examples. Solid lines are correct many-to-many alignment links.

NULL Substrings that do not need to be transliterated, typically white spaces and silent letters.

This work decides their distinction from the data; the choice of noise or NULL is learned from the bilingual string pairs[4].

4.2.2 Noise-aware Alignment with a Noise Assumption

This work introduces a *noise symbol* to handle partial noise in the many-to-many alignment model. (Htun et al., 2012) extended many-to-many alignment to sample-wise transliteration mining, but its noise model can only handle sample-wise noise and cannot distinguish partial noise. The partial noise is modeled in the CRP-based joint substring model.

Partial noise in transliteration data typically appears in compound words as mentioned earlier, because their counterparts consisting of two or more words may not be fully covered

[4]Most of the silent letters were included in many-to-many alignments and not aligned to NULL individually, in the experiments described later.

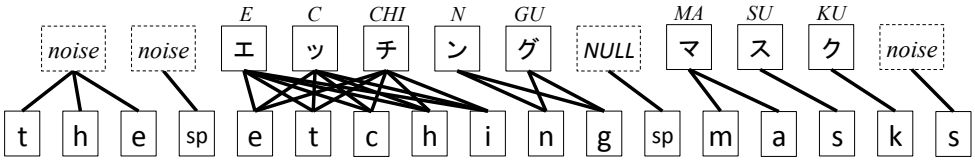
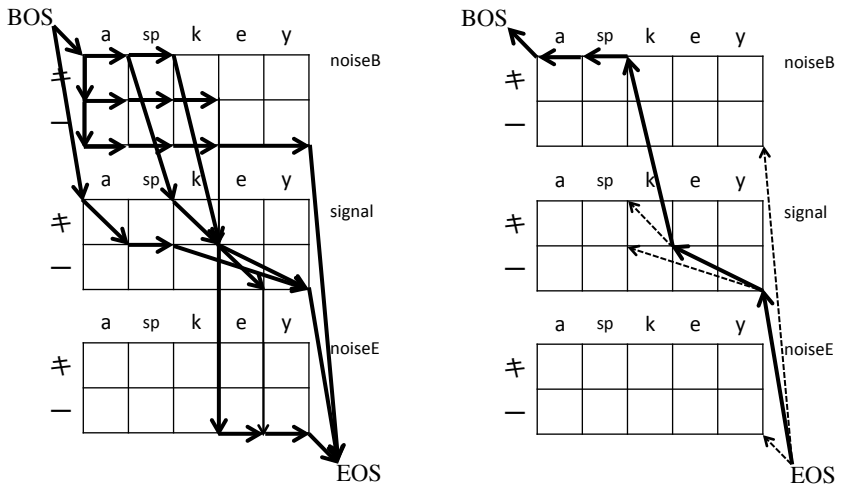


Figure 4.3: Example of many-to-many alignment with partial noise at the beginning and end. "*noise*" stands for the noise symbol, "*NULL*" stands for the zero-length substring, and "sp" indicates a white space.

in automatically extracted word and phrase pairs as shown in Figure 4.2(c). Another type of partial noise is derived from morphological differences caused by inflection, which usually appear at the sub-word level as prefixes and suffixes as shown in Figure 4.2(d) and 4.2(e). According to this intuition, I assume that partial noise appears in the beginning and/or end of transliteration data (I assume that noise appears at the beginning for sample-wise noise). This assumption derives a constraint between signal and noise parts that helps to avoid a welter of transliteration and non-transliteration parts. It also has a shortcoming in that it cannot handle noise in the middle (e.g., Figure 4.2(e)), but handling an arbitrary number of noise parts increases computational complexity and sparseness[5]. This paper is based on this simple assumption and reserves a more complex mid-noise problem as future work.

Figure 4.3 shows a partial noise example at both the beginning and end. This example is actually a correct translation but it includes noise in the sense of transliteration; the article "the" is wrongly included in the phrase pair (no articles are used in Japanese) and a plural noun "masks" is transliterated into "マスク"(mask). The noise symbols are treated as zero-length substrings in the model, which can be aligned to substrings same as other characters. The non-transliteration parts are aligned with noise symbols in the proposed model.

[5]Explicit modeling of the fixed number of noise parts (n) requires $2n$+1 different states in this paper's approach. Partial noise can be modeled by only signal and noise states but it may cause unexpected alignments with a welter of transliteration and non-transliteration parts.



(a) Forward filtering (b) Backward sampling

Figure 4.4: State-based FFBS for proposed model. (A few example paths are shown.)

4.2.3 State-based FFBS Extension

This work extends Finch's algorithm to the noise-aware model using a state-based calculation over the three states: the non-transliteration part at the beginning (noiseB), the transliteration part (signal), and the non-transliteration part at the end (noiseE). Substrings are aligned in either noiseB, signal, or noiseE state. In the noise states noiseB and noiseE, substrings are always aligned to the noise symbols. In the example of Figure 4.3, $\langle \textit{noise}, \text{the} \rangle$ and $\langle \textit{noise}, \text{sp} \rangle$ are aligned in noiseB, $\langle$エッチ, etchi$\rangle$, $\langle$ング, ng$\rangle$, $\langle \text{NULL}, \text{sp} \rangle$, $\langle$マ, ma$\rangle$, $\langle$ス, s$\rangle$, and $\langle$ク, k$\rangle$ are aligned in signal, $\langle \textit{noise}, \text{s} \rangle$ is aligned in noiseE. Here this work assumes that source-side noise is aligned first and target-side noise is aligned later in the noise states to avoid the repetitive counts of the same noise sequence, because co-segmentations $\langle \text{ab}, \textit{noise} \rangle \langle \textit{noise}, \text{xy} \rangle$ and $\langle \textit{noise}, \text{xy} \rangle \langle \text{ab}, \textit{noise} \rangle$ are equivalent.

The training framework of the proposed method is basically the same as Finch's one described in 4.1 and Algorithm 1, except for considering the partial noise and the three different states as shown in Figure 4.4. There are state transitions from noiseB to signal and signal to noiseE. Note that this work does not regard the state transitions as probabilistic events. This work introduces the noise states to constrain the appearance of the partial noise in the beginning and the end of bilingual string pairs as an extension of the Finch's

method.

Forward filtering

The state-based FFBS algorithm maintains three different forward probability matrices A^{noiseB}, A^{signal}, and A^{noiseE}, and corresponding forward probabilities α^{noiseB}, α^{signal}, and α^{noiseE}. $\alpha^{\mathsf{X}}(I, J)$ represents the marginal probability of a bilingual string pair $\langle s_1 ... s_I, t_1 ... t_J \rangle$ whose final substring pair is aligned in state X. The forward probabilities at the beginning in these states are initialized as $\alpha^{\mathsf{noiseB}}(0,0) = 1$, $\alpha^{\mathsf{signal}}(0,0) = 0$, and $\alpha^{\mathsf{noiseE}}(0,0) = 0$, because the beginning state is noiseB. The forward probabilities can be calculated efficiently by a dynamic programming using the matrices, same as the FFBS in Section 3. I explain the calculation of the forward probabilities in each state in the following.

The forward probabilities in state noiseB (α^{noiseB}) can be considered separately with two parts corresponding to source- and target-side noise. Here recall that the source-side partial noise is aligned first.

- For source-side noise as the downward arrows in noiseB state in Figure 4.4(a), the forward probability is the sum of the probabilities with different length of $\bar{\boldsymbol{s}}_k$ as follows:

$$\alpha^{\mathsf{noiseB}}(I, 0) = \sum_{\bar{\boldsymbol{s}}_k \in C(I, \boldsymbol{s})} p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \textit{noise} \rangle\right) \times \alpha^{\mathsf{noiseB}}\left(I - |\bar{\boldsymbol{s}}_k|, 0\right). \tag{4.10}$$

- For target-side noise as the rightward arrows in noiseB state in Figure 4.4(a), the forward probability is the sum of the probabilities with different length of $\bar{\boldsymbol{t}}_k$ as follows:

$$\alpha^{\mathsf{noiseB}}(I, J) = \sum_{\bar{\boldsymbol{t}}_k \in C(J, \boldsymbol{t})} p_{-m}\left(\langle \textit{noise}, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{noiseB}}\left(I, J - |\bar{\boldsymbol{t}}_k|\right). \tag{4.11}$$

The forward probabilities in state signal (α^{signal}) are almost the same as those of the original FFBS in Equation (4.7) but has two differences:

- state transitions from noiseB have to be considered (as illustrated by arrows from noiseB to signal in Figure 4.4(a)), and
- NULL is allowed in the many-to-many alignment, so either $\bar{\boldsymbol{s}}_k$ or $\bar{\boldsymbol{t}}_k$ can be a zero-length substring.

Thus, the calculation of α^{signal} is extended by two different previous states noiseB and signal, and by the NULL-aligned substrings as follows:

$$\begin{aligned}\alpha^{\mathsf{signal}}(I,J) = & \\ & \sum_{\bar{\boldsymbol{s}}_k \in C_N(I,\boldsymbol{s}),\ \bar{\boldsymbol{t}}_k \in C_N(J,\boldsymbol{t}),\ |\bar{\boldsymbol{s}}_k|+|\bar{\boldsymbol{t}}_k|>0} p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{noiseB}}\left(|\boldsymbol{s}| - |\bar{\boldsymbol{s}}_k|, |\boldsymbol{t}| - |\bar{\boldsymbol{t}}_k|\right) \\ & + \sum_{\bar{\boldsymbol{s}}_k \in C_N(I,\boldsymbol{s}),\ \bar{\boldsymbol{t}}_k \in C_N(J,\boldsymbol{t}),\ |\bar{\boldsymbol{s}}_k|+|\bar{\boldsymbol{t}}_k|>0} p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{signal}}\left(|\boldsymbol{s}| - |\bar{\boldsymbol{s}}_k|, |\boldsymbol{t}| - |\bar{\boldsymbol{t}}_k|\right) \end{aligned} \quad (4.12)$$

where $C_N(I, \boldsymbol{s})$ is a set of substrings same as $C(I, \boldsymbol{s})$ but includes a zero-length substring for NULL.

Finally, the forward probabilities in state noiseE (α_{noiseE}) have to consider the source-side noise and the target side noise, and further need to handle state transitions from signal.

- If the source-side does not reach its end ($I < |\boldsymbol{s}|$), the final substring must be the source-side noise. So only the source-side noise is considered as follows:

$$\begin{aligned}\alpha^{\mathsf{noiseE}}(I,J) = & \\ & \sum_{\bar{\boldsymbol{s}}_k \in C(I,\boldsymbol{s})} p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \textit{noise} \rangle\right) \times \alpha^{\mathsf{signal}}\left(I - |\bar{\boldsymbol{s}}_k|, J\right) \\ & + \sum_{\bar{\boldsymbol{s}}_k \in C(I,\boldsymbol{s})} p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \textit{noise} \rangle\right) \times \alpha^{\mathsf{noiseE}}\left(I - |\bar{\boldsymbol{s}}_k|, J\right). \end{aligned} \quad (4.13)$$

- Otherwise, the probabilities have to be summed up from the source-side noise towards the end of the source-side string (as illustrated by a downward arrow) in state noiseE, and the target-side noise along with the end of the source-side string (as illustrated by rightward arrows) in state noiseE:

$$\begin{aligned}\alpha^{\mathsf{noiseE}}(I = |\boldsymbol{s}|, j) = & \\ & \sum_{\bar{\boldsymbol{s}}_k \in C(I,\boldsymbol{s})} p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \textit{noise} \rangle\right) \times \alpha^{\mathsf{signal}}\left(|\boldsymbol{s}| - |\bar{\boldsymbol{s}}_k|, J\right) \\ & + \sum_{\bar{\boldsymbol{s}}_k \in C(I,\boldsymbol{s})} p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \textit{noise} \rangle\right) \times \alpha^{\mathsf{noiseE}}\left(|\boldsymbol{s}| - |\bar{\boldsymbol{s}}_k|, J\right) \\ & + \sum_{\bar{\boldsymbol{t}}_k \in C(J,\boldsymbol{t})} p_{-m}\left(\langle \textit{noise}, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{signal}}\left(|\boldsymbol{s}|, J - |\bar{\boldsymbol{t}}_k|\right) \\ & + \sum_{\bar{\boldsymbol{t}}_k \in C(J,\boldsymbol{t})} p_{-m}\left(\langle \textit{noise}, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{noiseE}}\left(|\boldsymbol{s}|, J - |\bar{\boldsymbol{t}}_k|\right). \end{aligned} \quad (4.14)$$

Backward sampling

The backward sampling is also extended to handle the three states. It firstly samples the ending state based on the forward probabilities at the end of the bilingual string pair: $\alpha^{\mathsf{noiseB}}(|\boldsymbol{s}|, |\boldsymbol{t}|)$, $\alpha^{\mathsf{signal}}(|\boldsymbol{s}|, |\boldsymbol{t}|)$, and $\alpha^{\mathsf{noiseE}}(|\boldsymbol{s}|, |\boldsymbol{t}|)$. Then it sets index (I, J) as $(|\boldsymbol{s}|, |\boldsymbol{t}|)$ and repeat the sampling of the bilingual substring pair $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$ towards the beginning represented by the indices $(0, 0)$, based on the posterior distributions of the substring pairs ending with the position represented by the indices (I, J). The indices are updated according to the sampled substrings. Here, the current state has to be maintained in the sampling. The backward sampling process samples a substring pair and its previous state at the same time to distinguish paths from different states, and update the current state to the sampled state.

Figure 4.4(b) shows an example of the backward sampling. This is basically similar to the original ones shown in Figure 4.1(b); the difference is that the proposed method handles three forward probability tables for the three states. At first, signal state is sampled as the ending state. Then substring pairs $\langle$ー$, ey\rangle$ and $\langle$キ$, k\rangle$ are sampled from signal state. When $\langle$キ$, k\rangle$ is sampled, there are two possibility of its previous state, noiseB and signal, as shown in the figure. Here $\langle$キ$, k\rangle$ is sampled from noiseB, so the current state have changed to noiseB. Finally, $\langle noise, \mathsf{sp}\rangle$ and $\langle noise, a\rangle$ are sampled from noiseB state. I explain the calculation of the posterior probabilities of substring pairs to be sampled in the following.

If the current state is noiseB, the previous state is also noiseB so state transitions are not needed. The cases of source- and target-side noise are distinguished in the forward probability calculation.

- If J is equal to zero, preceding source-side noise $\bar{\boldsymbol{s}}_k$ is sampled among $C(I, \boldsymbol{s})$ using

$$p\left(\langle \bar{\boldsymbol{s}}_k, \mathit{noise}^{\mathsf{noiseB}} \rangle \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \mathit{noise} \rangle\right) \times \alpha^{\mathsf{noiseB}}\left(I - |\bar{\boldsymbol{s}}_k|, 0\right). \quad (4.15)$$

- Otherwise, target-side noise $\bar{\boldsymbol{t}}_k$ is sampled among $C(J, \boldsymbol{t})$ using

$$p\left(\langle \mathit{noise}, \bar{\boldsymbol{t}}_k \rangle^{\mathsf{noiseB}} \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \mathit{noise}, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{noiseB}}\left(I, J - |\bar{\boldsymbol{t}}_k|\right). \quad (4.16)$$

Here, the notation $p\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle^{\mathsf{X}} \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right)$ means the posterior probability of the substring pair $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$ whose previous state is X.

If the current state is signal, the previous state is either noiseB or signal. The substring pairs are distinguished based on their previous states, and sample a substring pair $\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle$ among $C_N(I, \boldsymbol{s})$ and $C_N(J, \boldsymbol{t})$ based on the following posterior probabilities:

$$p\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle^{\mathsf{noiseB}} \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{noiseB}}\left(I - |\bar{\boldsymbol{s}}_k|, J - |\bar{\boldsymbol{t}}_k|\right), \text{and} \quad (4.17)$$

$$p\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle^{\mathsf{signal}} \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{signal}}\left(I - |\bar{\boldsymbol{s}}_k|, J - |\bar{\boldsymbol{t}}_k|\right). \quad (4.18)$$

If a substring pair is sampled from the distribution of Equation (4.17), the current state is changed to noiseB.

Finally, if the current state is noiseE, its preceding state is either signal or noiseE. The cases of source- and target-side noise, and two different previous states are distinguished.

- If the current source-side index I is smaller than the length of the source-side string $|\boldsymbol{s}|$, there must be no preceding target-side noise. Source-side noise is sampled considering its previous state between signal or noiseE based on the following distributions:

$$p\left(\langle \bar{\boldsymbol{s}}_k, \mathit{noise}^{\mathsf{signal}} \rangle \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \mathit{noise} \rangle\right) \times \alpha^{\mathsf{signal}}\left(I - |\bar{\boldsymbol{s}}_k|, J\right), \text{and} \quad (4.19)$$

$$p\left(\langle \bar{\boldsymbol{s}}_k, \mathit{noise} \rangle^{\mathsf{noiseE}} \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \bar{\boldsymbol{s}}_k, \mathit{noise} \rangle\right) \times \alpha^{\mathsf{noiseE}}\left(I - |\bar{\boldsymbol{s}}_k|, J\right). \quad (4.20)$$

- Otherwise, the source- or target-side noise can be sampled using the following probabilities together with the source-side noise probabilities of Equations (4.19) and (4.20), considering its previous state signal and noiseE:

$$p\left(\langle \mathit{noise}, \bar{\boldsymbol{t}}_k \rangle^{\mathsf{signal}} \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \mathit{noise}, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{signal}}\left(I, J - |\bar{\boldsymbol{t}}_k|\right), \text{and} \quad (4.21)$$

$$p\left(\langle \mathit{noise}, \bar{\boldsymbol{t}}_k \rangle^{\mathsf{noiseE}} \mid \langle \boldsymbol{s}, \boldsymbol{t} \rangle\right) \propto p_{-m}\left(\langle \mathit{noise}, \bar{\boldsymbol{t}}_k \rangle\right) \times \alpha^{\mathsf{noiseE}}\left(I, J - |\bar{\boldsymbol{t}}_k|\right). \quad (4.22)$$

If a substring pair is sampled from the distribution of Equation (4.19) or (4.21), the current state is changed to signal.

The computational cost with this algorithm is increased almost three-fold compared with that of Finch and Sumita (2010), because it handles three different states.

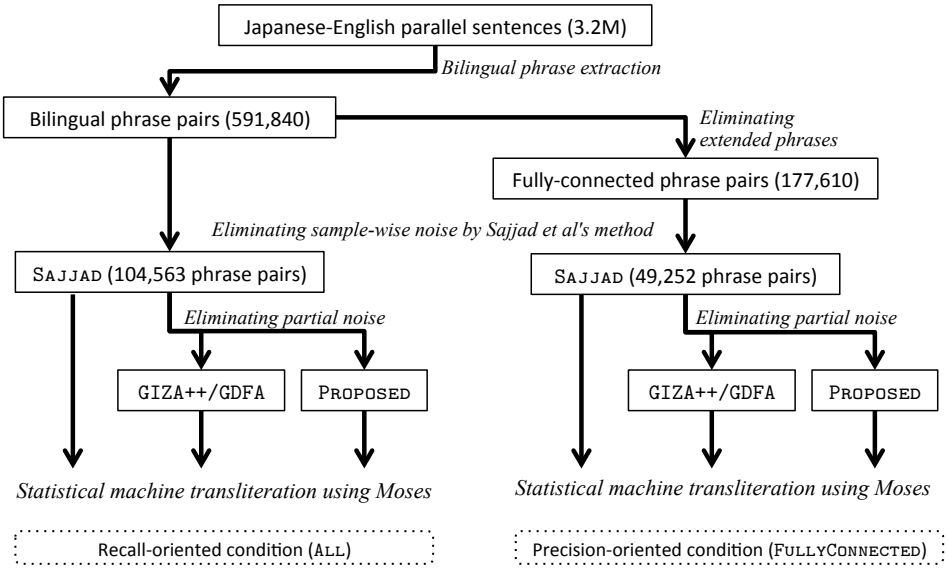


Figure 4.5: Workflow of the transliteration bootstrapping experiments.

4.3 Experiments

Japanese-to-English transliteration experiments were conducted, in which the proposed method was compared with the conventional sample-wise method in bootstrapping statistical machine transliteration employing a Japanese-to-English patent translation dataset. The experiments focused on *katakana* words in Japanese that are usually used for the transliteration of foreign words, and these *katakana* words were back-transliterated into the original English words. Note that the problem of back-transliteration generally has unique answers because most transliterated *katakana* words have unique corresponding foreign words except for some homonyms, while transliteration is basically ambiguous due to non-unique sound-to-character mappings.

The workflow of the experiments is illustrated in Figure 4.5, and the following sections give its detailed explanation.

4.3.1 Setup

For the transliteration bootstrapping, bilingual phrase pairs with a maximum length of seven words were extracted from 3.2M parallel sentences in NTCIR-10 PatentMT dataset (Goto et

al., 2013)[6], by a standard training procedure for phrase-based SMT (Koehn et al., 2003) with GIZA++[7] and Moses[8]. Japanese and English sentences were tokenized using MeCab[9] and `tokenizer.perl` (included in Moses), respectively. 591,840 unique phrase pairs whose Japanese side was written in katakana only[10] were obtained. Here, some bilingual phrases include words without appropriate counterparts (called NULL-aligned words). The bilingual phrases were extracted from bilingual corpora with automatic word alignment; but in the Japanese-English case, English articles and prepositions are sometimes not aligned with any Japanese words because they lack Japanese counterparts. These unaligned words are needed for actual translation and typical phrase-based machine translation uses *extended* phrases in which such unaligned words are included[11]. These extended phrases can extract complete phrasal correspondences of compound words but also incorporate more noise in the phrase pairs. Thus, in the experiments, two conditions were compared to take this problem into account:

- Recall-oriented condition (All): all the bilingual phrases (591,840) are used to extract more transliteration substring pairs; and
- Precision-oriented condition (FullyAligned): only fully aligned phrases without unaligned words (177,610) to exclude more word-level noise.

The method proposed by Sajjad et al. (2012) was first used iteratively on these bilingual phrases and sample-wise non-transliteration pairs were eliminated, until the number of pairs converged. Finally 104,563 *katakana*-English pairs were obtained from All after 10 iterations, and 49,252 pairs from FullyAligned after 8 iterations. They were the *baseline transliteration training set* mined by the sample-wise method. Sajjad et al.'s method was used as a preprocessing technique for filtering sample-wise noise. The proposed method is also capable of this, but it takes much more training time for all phrase table entries. This

[6] http://research.nii.ac.jp/ntcir/data/data-en.html

[7] https://code.google.com/p/giza-pp/

[8] http://www.statmt.org/moses/

[9] http://code.google.com/p/mecab/

[10] This katakana-based filtering is a language dependent heuristic for choosing potential transliteration candidates, because transliterations in Japanese are usually written in katakana.

[11] For more details, please refer to the book of SMT (Koehn, 2010).

work focuses on the partial noise problem in the experiments that was not addressed by Sajjad *et al*.'s method.

Then the proposed method was applied to the baseline transliteration training set with 30 sampling iterations (empirically chosen for the convergence of the likelihood) and obtained character alignment results with partial noise identification. Here, the hyperparameters, α, λ_s, and λ_t, were optimized using a held-out set of 2,000 *katakana*-English pairs that were randomly chosen from a general-domain bilingual dictionary[12]. The hyperparameter optimization was based on transliteration F-score values on the held-out set with the following α values 0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, and the following λs values 1, 2, 3, 4, 5. Finally $\alpha = 0.1$ and $\lambda = 3$ were used for All condition, and $\alpha = 0.02$ and $\lambda = 2$ were used for FullyAligned condition.

4.3.2 Evaluation of Partial Noise Identification

First the extent to which the proposed method identified partial noise was evaluated using 4,000 *katakana*-English pairs randomly chosen from the baseline transliteration training set of All. The partial noise of the 4,000 pairs was manually annotated by an annotator. 251 of 34,547 (0.73%) *katakana* characters and 3,462 of 52,872 (6.5%) English characters were annotated as noise, including three sample-wise noise pairs that were not identified by Sajjad et al.'s method. In the evaluation in the FullyAligned condition, 1,524 pairs were used, which were also included in the baseline transliteration training set of FullyAligned, out of the 4,000 annotated pairs. In the evaluation, the proposed method (Proposed) was compared with a baseline (GIZA++/GDFA) where unaligned characters using bilingual phrase extraction heuristics called *grow-diag-final-and* over bidirectional GIZA++ alignment[13] were regarded as noise.

[12] the parameters of the spelling model λ_s, and λ_t can be learned from co-segmentation samples (because they are equals to expectations of substring lengths according to a Poisson distribution), but constant values were used same as (Finch and Sumita, 2010).

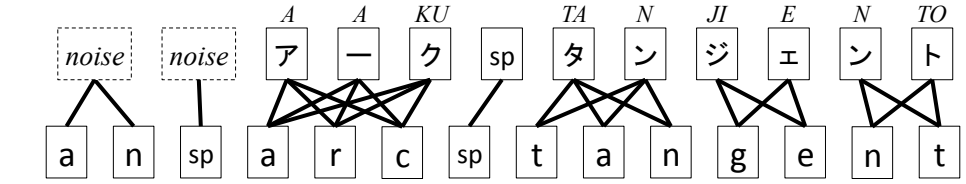
[13] Word alignments based on IBM models (Brown et al., 1993) and HMM alignment model (Vogel et al., 1996) used in GIZA++ are many-to-one, so recent phrase-based statistical machine translation combines these many-to-one alignments with one-to-many alignments in the reverse direction to obtain many-to-many alignments for bilingual phrases. *grow-diag-final-and* is a commonly-used heuristic alignment combination method. For more details, please see (Koehn, 2010).

Table 4.1: Precision(%), recall(%), and F-measure(%) of noise identification by the proposed method (Proposed) and IBM-model-based baseline (GIZA++/GDFA) using 4,000 (All) and 1,524 (FullyAligned) noise annotated phrase pairs.

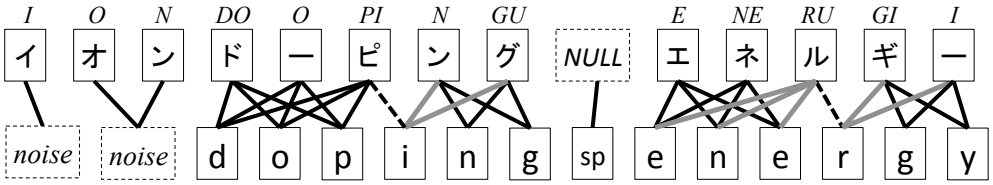
Phrases	Language	Method	Precision(%)	Recall(%)	F-measure(%)
All	Japanese	GIZA++/GDFA	56.4	8.09	14.2
		Proposed	83.6	23.7	37.0
	English	GIZA++/GDFA	42.3	2.28	4.32
		Proposed	85.8	5.24	65.1
FullyAligned	Japanese	GIZA++/GDFA	51.0	6.89	12.1
		Proposed	72.2	19.3	30.4
	English	GIZA++/GDFA	7.14	0.122	0.241
		Proposed	58.0	22.3	32.2

Table 4.1 shows precision, recall, and F-measure values for noise identification both in Japanese and English for different phrase extraction conditions. GIZA++/GDFA was clearly worse than Proposed. This suggests that NULL alignments in IBM models are not appropriate for identifying partial noise. With respect to the difference between All and FullyAligned, the performance of Proposed for FullyAligned was much worse than that for All. One possible reason for this is the noise on the English side in the All phrases, which appeared as articles and prepositions and that can be easily eliminated as word-level noise.

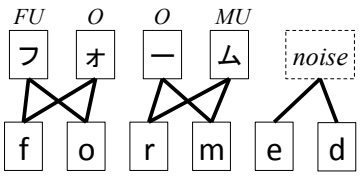
Figure 4.6 shows examples of the alignment results in the training data. As expected, partial noise both in Japanese and English was identified correctly in (a), (b), and (c). There were some alignment errors in the signal part in (b), in which characters in boundary positions were aligned incorrectly with adjacent substrings. These alignment errors did not directly degrade the partial noise identification but they may have a negative effect the overall alignment performance in the sampling-based optimization. (d) is a negative example in which partial noise was incorrectly aligned. (c) and (d) have similar partial noise in their English word endings, but it could not be identified in (d).



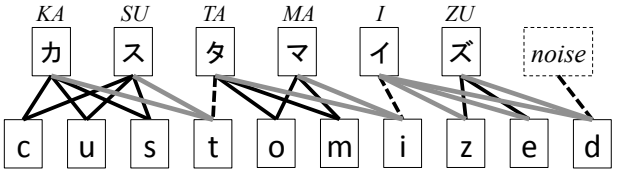
(a) Correctly aligned



(b) Some alignment errors in transliteration part



(c) Correctly aligned



(d) Errors in partial noise

Figure 4.6: Examples of noise-aware many-to-many alignment in the training data. ϕ indicates a zero-length substring. Bold gray lines show incorrect alignments, and dashed lines mean their corrections.

4.3.3 Evaluation of Transliteration Accuracy

Next the transliteration accuracy by the use of the mined transliteration pairs was evaluated. Statistical machine transliteration was implemented as character-based statistical machine translation with Moses, using a character-based 7-gram language model trained on 300M English patent sentences. The test set was the top 1000 unknown (in the Japanese-to-English translation model) *katakana* words appearing in 400M Japanese patent sentences. They covered 15.5% of all unknown *katakana* words and 8.8% of all unknown words (excluding numbers); that is, more than half of the unknown words were *katakana* words. The problem itself was Japanese-to-English back-transliteration as described above.

Three different training data of the transliteration were compared: Sajjad et al's method (Sajjad; namely the baseline transliteration training set), GIZA++/GDFA and Proposed. Table 4.2 shows the data statistics obtained after eliminating noise with them. Recall that

Table 4.2: Statistics of the transliteration training sets after eliminating sample-wise and partial noise. The sample-wise noise was eliminated by Sajjad *et al*'s method (baseline transliteration training set), and partial noise was further eliminated by GIZA++ and grow-diag-final-and heuristics (GIZA++/GDFA) and the proposed method (Proposed).

Phrases	Method	#pairs	#Ja chars.	#En chars.
All	Sajjad	104,563	899,080	1,372,993
	+GIZA++/GDFA	104,563	894,985	1,323,365
	+Proposed	104,561	893,366	1,317,256
FullyAligned	Sajjad	49,252	378,280	610,831
	+GIZA++/GDFA	49,252	376,552	590,409
	+Proposed	49,245	372,970	587,514

GIZA++/GDFA and Proposed were applied to Sajjad results as described above and the number of phrase pairs and characters by GIZA++/GDFA and Proposed were lower than those of Sajjad. The training procedure of statistical machine transliteration was a standard Moses approach: GIZA++-based alignment, grow-diag-final-and alignment symmetrization and phrase extraction with a maximum phrase length of seven characters. Note that the Bayesian many-to-many alignment was ignored in bilingual phrase extraction and phrases were re-aligned with GIZA++ and grow-diag-final-and heuristic, because the effect of partial noise elimination on statistical machine transliteration was investigated in the same training condition. The use of the many-to-many alignment as bilingual phrases as Finch and Sumita (2010), called Proposed-Joint, was also tested following their agglomeration heuristic to include longer substring pairs by:

1. generating many-to-many word alignment, in which all possible word alignments link in many-to-many correspondences (e.g., 0-0 0-1 0-2 1-0 1-1 1-2 for $\langle$コ ン, $com\rangle$),
2. running phrase extraction and scoring the same as with standard Moses training.

This procedure extracts longer phrases that satisfy the many-to-many alignment constraints than the simple use of extracted joint substring pairs as phrases.

ACC (sample-wise accuracy) was used as the main evaluation metric, the sample-

wise accuracy in the back-transliteration of *katakana* words compared with original English words. Two additional metrics were used for character-wise evaluation: F-score, and BLEU_c. F-score is a character-wise F-measure-like score (Li et al., 2010). BLEU_c is BLEU (Papineni et al., 2002) at the character level with n=4. Table 4.3 shows the transliteration evaluation results.

Proposed achieved ACC of 63% (16% relative error reduction compared with Sajjad) using All phrases and 65% (8% relative error reduction compared with Sajjad) using FullyAligned phrases. It also showed better character-wise performance in F-score and BLEU_c. These improvements clearly exhibited the advantage of the proposed method over sample-wise mining. Recall that Sajjad and Proposed had a small difference in their training data size as shown in Table 4.2. In contrast, GIZA++/GDFA was based on a similar-sized training set but produced much worse ACC results than Sajjad. Proposed further eliminated partial noise from the sample-wise mined results and achieved the best back-transliteration performance. These results suggest that the partial noise can hurt transliteration models and the proposed approach actually worked in transliteration bootstrapping.

Proposed-Joint performed similarly to Proposed, although many-to-many substring alignment was expected to improve transliteration as reported by (Finch and Sumita, 2010). The difference may be due to the difference in coverage of the phrase tables; Proposed-Joint retained relatively long substrings caused by the many-to-many alignment constraints in contrast to the less-constrained grow-diag-final-and alignments in Proposed. Since the training data in the bootstrapping experiments contained many similar phrases unlike the dictionary-based data in Finch and Sumita (2010), the Proposed-Joint phrase table may have limited coverage owing to its long and sparse substring pairs with large probabilities even if the many-to-many alignment was good. This sparseness problem is beyond the scope of this paper and worth further study.

Some transliteration examples are shown in Table 4.4. The first two examples show a typical advantage of Proposed. Sajjad generated noise (ester“s” and “an” armonk) due to partial noise in the training data. In the next example, both methods generated wrong transliterations. The transliteration results for the word “protoplast” include unnecessary suffixes “-ing” that should be aligned to the noise symbol by the proposed method. This is mainly due to the low recall in noise identification, as shown in Table 4.1. In the last three

Table 4.3: Japanese-to-English transliteration results for the top 1000 unknown katakana words. ACC and F-score stand for those used in NEWS workshop, $BLEU_c$ is character-wise BLEU. Values shown in **bold** represent the best values for the same phrase extraction condition.

Phrases	Method	ACC	F-score	$BLEU_c$
All	Sajjad	0.56	0.929	0.864
	+GIZA++/GDFA	0.47	0.929	0.850
	+Proposed	**0.63**	**0.946**	**0.897**
	Proposed-Joint	**0.63**	0.943	0.888
FullyAligned	Sajjad	0.62	0.943	0.887
	+GIZA++/GDFA	0.49	0.932	0.851
	+Proposed	0.65	**0.949**	**0.901**
	+Proposed-Joint	**0.66**	0.947	0.899

Table 4.4: Transliteration examples by Sajjad and Proposed in FullyAligned condition.

Katakana	Reference	Sajjad	Proposed
ロジンエステル (RO JI N E SU TE RU)	rosin ester	rosin esters	rosin ester
アーモンク (A A MO N KU)	armonk	an armonk	armonk
プロトプラスト (PU RO TO PU RA SU TO)	protoplast	protoplasting	protoplasting
サラダ (SA RA DA)	salad	salader	salader
ローヤル (RO O YA RU)	royal	low ear r	low al
ローダニン (RO O DA NI N)	rhodanine	loadenine	loader nin

examples, the transliteration results are not appropriate as English words, although they may have similar pronunciations to their reference words. Longer bilingual phrases and n-gram language models are expected to choose more consistent hypotheses, but they sometimes fail to keep word-level consistency and then generate inappropriate character sequences especially for rare words. This consistency problem in statistical machine transliteration warrants further study, not only in segmental alignment but also in context-rich modeling and decoding.

4.4 Related Work

Machine transliteration is often treated as a sub-problem of machine translation and cross-lingual information retrieval for handling unknown names and terms. There have been many previous studies on machine transliteration between various languages, as described in a previous survey (Karimi et al., 2011).

This work relates to a technology in the field of machine transliteration called *transliteration mining* or *transliteration extraction*, which aims to find transliteration pairs from parallel, comparable, or even independent bilingual resources. A typical task in transliteration mining involves finding transliteration pairs at the word level from Wikipedia Inter-Language Link data, as in a shared task in the 2010 Named Entity Workshop (Kumaran et al., 2010). The problem in such a task is classifying transliteration pair hypotheses into transliteration and non-transliteration. Fukunishi et al. (2013) applied Finch et al.'s many-to-many alignment model to this classification task, using forced-aligned substring pairs as features for support vector machine-based classification. This work aims to find *segmental* transliteration pairs excluding partial noise, not sample-wise pairs as in previous studies, as the first work on this kind of problem.

Technically this work is based on transliteration bootstrapping (Sajjad et al., 2012) and many-to-many character alignment (Finch and Sumita, 2010), and extends them to this problem. But the proposed approach is not limited to the current implementation; transliteration candidates can be explored from comparable or independent bilingual resources by other transliteration mining technologies (Al-Onaizan and Knight, 2002; Lam et al., 2004); other transliteration alignment methods (such as the EM-based approach (Kubo et al., 2011)) can be extended to partial noise.

4.5 Conclusion

This work proposed a noise-aware many-to-many alignment model that can distinguish partial noise in transliteration pairs for bootstrapping a statistical machine transliteration model from automatically extracted phrase pairs. The model and training algorithm are straightforward extensions of those described by Finch and Sumita (2010). The proposed

method was proved effective in partial noise identification and transliteration bootstrapping in experiments with Japanese-to-English patent documents.

Chapter 5

Syntax-based Post-ordering for Efficient Reordering

There are two main problems in SMT: lexical (word and phrasal) translation and reordering. The standard SMT approach solves these problems jointly in an integrated search with a limited reordering distance to reduce search complexity. However, this limit must be set at a large value for languages requiring extremely long distance reordering (e.g., Japanese-English), thereby restricting its usefulness as a method for increasing decoding speed. This work tackles the difficulty of long distance reordering in Japanese-to-English translation.

One promising approach is to employ a two-step framework in which lexical translation and reordering are isolated explicitly, in contrast to the integrated search. In a sense, this two-step framework provides another alternative besides the distortion limit for simplifying the search space. Most previous two-step technologies use *pre-ordering* to reorder source language words in the target language word order *before* lexical translation (Xia and McCord, 2004; Collins et al., 2005; Costa-jussà and Fonollosa, 2006; Li et al., 2007; Tromble and Eisner, 2009; Xu et al., 2009; Hong et al., 2009; Genzel, 2010). Currently pre-ordering works very effectively in English-to-Japanese translation utilizing syntactic parsing in English (Isozaki et al., 2010b; Isozaki et al., 2012), while that in Japanese-to-English translation (Katz-Brown and Collins, 2008)(Kondo et al., 2011; Hoshino et al., 2013b) is much less effective, possibly due to an asymmetry in reordering.

There is another two-step method that employs reordering *after* lexical translation using

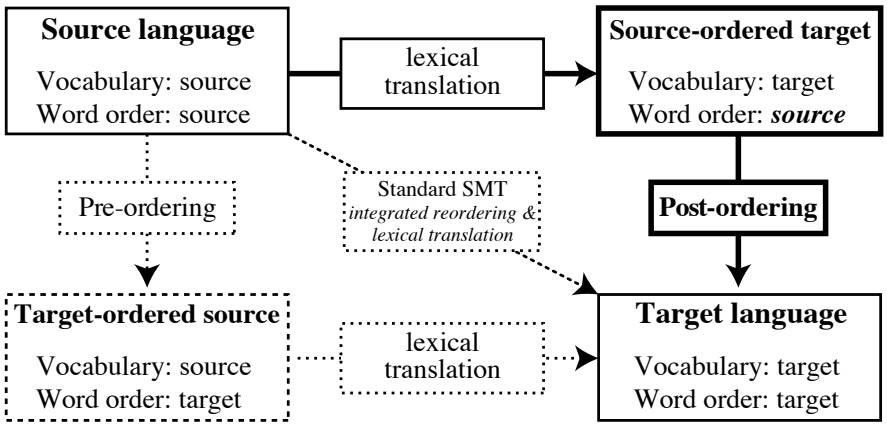


Figure 5.1: Translation directions of standard, pre-ordering and post-ordering SMT approaches.

an intermediate source-ordered target language (Bangalore and Riccardi, 2000; Matusov et al., 2005; Na et al., 2009). This approach is called *post-ordering* as opposed to pre-ordering (as illustrated in Figure 5.1) . Previous post-ordering methods employ simple reordering models that are insufficient for Japanese-to-English translation with long distance reordering. This work proposes a novel syntactic post-ordering method focusing on two problems:

- how to generate the intermediate language, and
- how to solve the post-ordering with long distance reordering.

For the first problem, there is an important advantage that Japanese-ordered English can be induced from English using the existing accurate English-to-Japanese syntactic pre-ordering method (Isozaki et al., 2010b; Isozaki et al., 2012). This work tackle the second problem as another SMT problem by accurate syntax-based SMT with target language syntax. The two-step SMT with the proposed method provides a viable alternative to one-step syntax-based SMT, which is generally very accurate even with long distance reordering but very slow in practice. In Japanese-to-English patent translation experiments the proposed method achieves six times faster decoding speed than a baseline, with comparable translation accuracy.

5.1 Two-step Statistical Machine Translation with Post-ordering

As illustrated in Figure 5.1, a standard SMT approach provides a combined solution to the lexical translation and reordering problems. Syntax-based SMT (Galley et al., 2004; Chiang, 2007; Zollmann and Venugopal, 2006) solves them jointly using synchronous grammar rules. Phrase-based SMT (Koehn et al., 2003; Tillmann, 2004) also solves them jointly although it is *implicitly* split by using isolated phrasal translation and reordering models. In contrast, two-step approaches *explicitly* split them using intermediate languages. This work focuses on the two-step approach with post-ordering using a source-ordered target language as its intermediate language.

The use of an intermediate "source-ordered target" language was proposed by Bangalore and Riccardi (2000), for tightly integrated spoken language translation with weighted finite state transducers (WFSTs). Their method determines dependency-based reordering in source-ordered target sentences using a finite-state parsing model, which can be trained using bilingual corpora with automatic word alignment. Its finite-state model only uses word surfaces as constraints and is not sufficient to capture syntactic structure and reordering for long sentences. Matusov et al. (2005) and Bangalore et al. (2007) employed simple permutation models with reordering limits; they focused more on lexical translation problems and were less concerned about long distance reordering. Recently Goto et al. (2012) followed our work by extending parsing-based post-ordering (Bangalore and Riccardi, 2000), using tree-to-tree correspondence between English and Japanese-ordered English.

5.2 Proposed Method

This work propose a novel post-ordering method that works efficiently even with long distance reordering in Japanese-to-English translation, by employing reordering rules used for English-to-Japanese translation. Its key ideas are:

- the Japanese-ordered English can be induced from English with reordering for "the reverse direction", English-to-Japanese

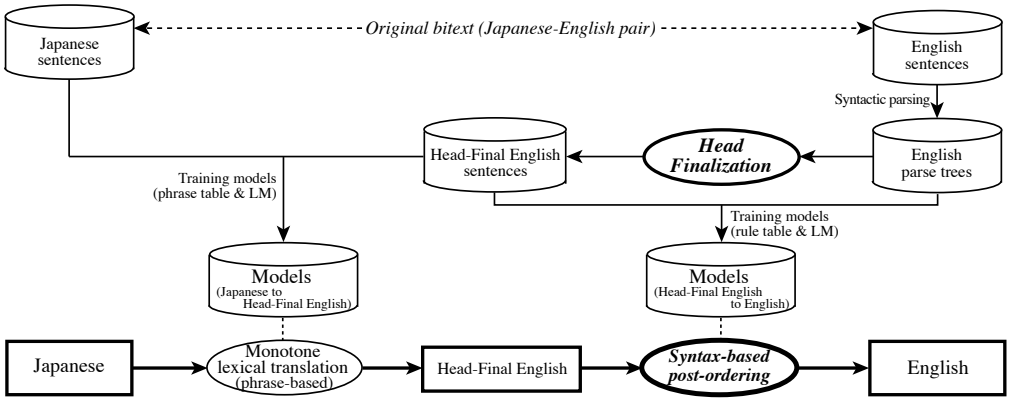


Figure 5.2: Japanese-to-English SMT workflow with proposed method.

- the post-ordering from the source-ordered target language to the target language can be regarded as an SMT problem

The proposed method uses *Head-Final English* (HFE) (Isozaki et al., 2010b) as the Japanese-ordered English, which is very effective in English-to-Japanese translation with pre-ordering.

Figure 5.2 shows an overall SMT workflow with the proposed method. The two-step SMT can be realized by employing two isolated SMT processes: Japanese-to-HFE lexical translation and HFE-to-English post-ordering. Models for these problems are trained using a trilingual corpus: the original bilingual corpus of Japanese and English, and HFE induced from English by Head Finalization. The first step is undertaken using a standard monotone phrase-based SMT and the second step is undertaken using a syntax-based SMT.

5.2.1 First Step: Lexical Translation from Japanese to Head-Final English

The proposed method solves the lexical translation problem as a monotone phrase-based SMT problem from Japanese to HFE. HFE was proposed in an English-to-Japanese pre-ordering study based on the head-final characteristics of Japanese; syntactic head words are almost always located after their modifier words in Japanese. Figure 5.3 shows an example of HFE together with a parse tree, for an English sentence:

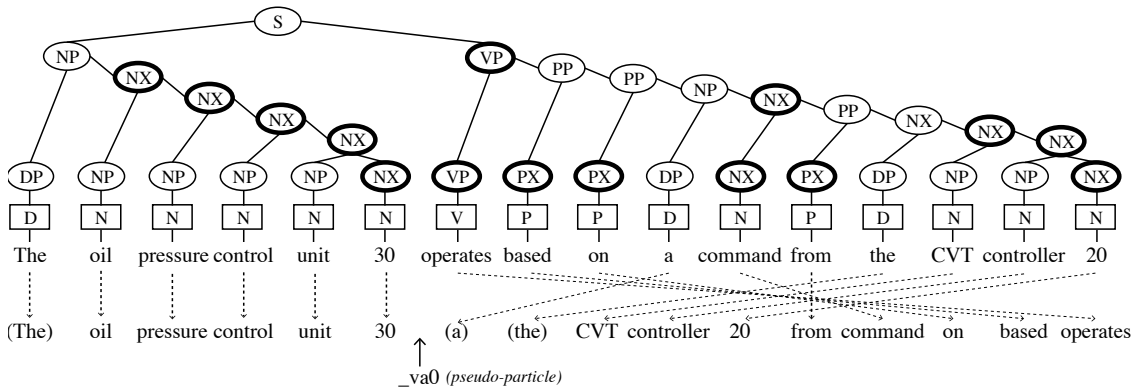


Figure 5.3: An example parse tree and corresponding HFE sentence. Nodes with a bold outline represent syntactic heads for each tree node. The articles "the" and "a" are eliminated by the rules, and a *pseudo*-particle "_va0" is inserted after the subject "The oil pressure control unit 30".

> *The oil pressure control unit 30 operates based on a command from the CVT controller 20.*

HFE is different from English as follows (see (Isozaki et al., 2010b; Isozaki et al., 2012) for details).

(1) Syntactic head words (represented with bold ovals in Figure 5.3) are located at the end of their siblings (except for coordination).

(2) Plural nouns (POS: `NNS`) are replaced with singular nouns.

(3) Articles "a", "an", and "the" are eliminated.

(4) *Pseudo*-particles are inserted immediately after verb arguments[1]:

- `_va0` for subjects of the sentence head verb
- `_va1` for subjects of other verbs
- `_va2` for objects of verbs

[1] A syntactic perspective is adopted for the subject-object relation, although it should be semantically swapped for passive voice verbs.

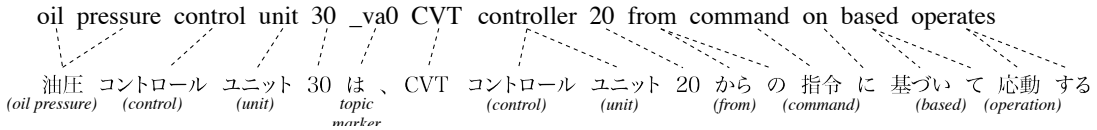


Figure 5.4: Word alignments between HFE and Japanese.

[Input Japanese sentence]
第 1 f θ レンズ 17 は 、第 1 レンズ 22 と 第 2 レンズ 23 と から 構成 される 。

↓ *Monotone lexical translation (Japanese-to-HFE)*

[Intermediate head-final English sentence]
first f θ lens 17 _va0 first lens 22 and second lens 23 of composed is .

↓ *Syntax-based post-ordering with long distance reordering (HFE-to-English)*

[Output English sentence]
the first f θ lens 17 is composed of a first lens 22 and the second lens 23 .

[Reference]
the first f θ lens 17 is composed of a first lens 22 and a second lens 23 .

Figure 5.5: Example of two-stage translation in post-ordering approach.

Point (1) above relates to the head-final word order, and the others are intended to achieve better word-level alignment between Japanese and HFE by bridging morpho-syntactic gaps (extends Korean-English heuristics (Hong et al., 2009) to Japanese-English). As shown in Figure 5.4, the HFE words have monotonic word correspondences with the Japanese words. This monotonicity makes Japanese-to-HFE lexical translation monotonic, as shown in the upper part of Figure 5.5.

5.2.2 Second Step: Syntax-based Post-ordering from Head-Final English to English

The proposed method tackles the post-ordering problem as another SMT problem from HFE to English. In the proposed approach, this problem can be regarded as the inverse of the problem of Head Finalization from English to HFE. The syntax-based SMT is applied

to this problem for the following reasons:

- The order of words in English depends strongly on their syntactic roles and syntactic information in English is expected to help the post-ordering.
- Many accurate English syntactic parsers have been developed.
- Syntax-based SMT works well in long distance reordering, although it requires long decoding time (as presented later in Section 5.3).

Since English syntactic parsing is used to obtain HFE, a parallel corpus of HFE strings and English parse trees can be obtained from English sentences. Here, when training translation rules, word alignments between HFE and English are *obvious* and their phrasal correspondence can easily be identified[2]. It is also worth noting that HFE and English have certain differences as regards plural nouns and articles of English, and pseudo-particles of HFE. These differences are included in the translation rule table and the post-ordering step corrects them.

In the intermediate lexical translation result in Figure 5.5, the English verb phrase "is composed of" is reversed and located at the end of the sentence as in Japanese. There are also no articles "a" and "the", but there is a pseudo-particle `_va0`. The word order is corrected by the post-ordering as an HFE-to-English translation. Here articles are inserted and the pseudo-particle is eliminated.

5.2.3 Time Complexity

The time complexity of syntax-based SMT by a CKY-based decoding with binarized grammars is $O(n^3)$, where n is the number of input words (Chiang, 2007). This can be reduced by introducing a reordering limit (maximum word span in the CKY-based decoding) but it is not suitable for long distance reordering. On the other hand, the time complexity of the *monotone* lexical translation step with Moses-like stack decoding is $O(n)$ (Koehn, 2010) and the syntax-based post-ordering also has the time complexity of $O(n^3)$. Thus the time complexity of the proposed method is theoretically equivalent to that of the original syntax-based SMT.

[2]This is a similar methodology to the "reordering tuples" presented by Costa-jussà and Fonollosa (2006).

Here, there is an important difference between Japanese-to-English translation and HFE-to-English post-ordering as regards their translation ambiguity. Japanese-to-English translation has to explore many competing hypotheses in lexical translation with a large number of translation rules. As a result, the actual computational cost of the standard syntax-based SMT is large to handle large lexical translation ambiguity. In contrast, HFE-to-English translation only tackles the reordering problem and has very small lexical translation ambiguity (only with plural nouns, articles, and pseudo-particles). Thus, the number of translation rules is expected to be small and the proposed method runs efficiently with a small translation ambiguities.

5.2.4 Asymmetry in Pre-ordering between English-to-Japanese and Japanese-to-English

Here it is worth noting the asymmetry that exists in pre-ordering between English-to-Japanese and Japanese-to-English translation.

As shown in the example in Figure 5.4, HFE aligns almost monotonically with English. Japanese is a typical head-final language, so that English syntactic head words can be reordered after their modifiers regardless of their syntactic roles. This pre-ordering process has little uncertainty and can determine Japanese word order systematically by using only English-side information. However, English is primarily a head-initial language but also allows a head-final order such as subject-verb and adjective-noun. This uncertainty makes it difficult to reorder Japanese words in English word order. In the example sentence in Figure 5.4, Subject-verb and adjective-noun relations have to be distinguished to induce the English-ordered Japanese as shown in Figure 5.6, based on a chunk-based dependency structure (commonly used in Japanese). Distinguishing head-final from head-initial relations is difficult even with state-of-the-art Japanese parsers, and therefore it is difficult to realize Japanese-to-English pre-ordering as in English-to-Japanese. In contrast, the proposed method provides one solution for this asymmetry issue by utilizing “pre-ordering in easier direction”.

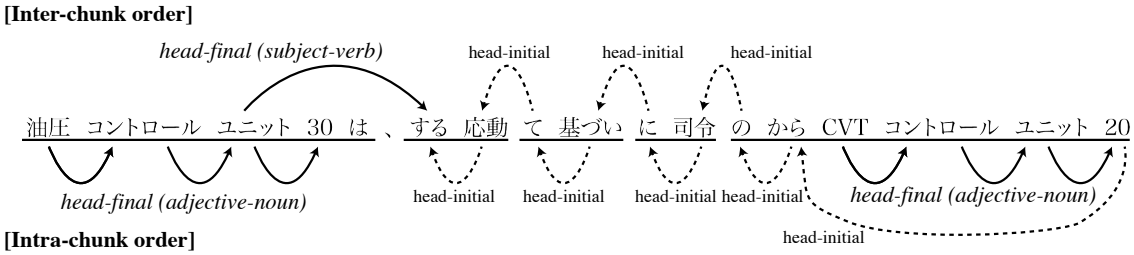


Figure 5.6: Mixture of head-final (solid lines) and head-initial (dashed lines) order in English-ordered Japanese (artificial reordering example).

5.3 Experiments

The efficiency of the two-step SMT approach with the proposed method was investigated by undertaking the following Japanese-to-English translation experiment.

5.3.1 Setup

The NTCIR-9 PatentMT (Goto et al., 2011) English and Japanese dataset were used for this experiment, with its development set of 2,000 sentence pairs as the development and test sets of 1,000 sentences each dividing the original development set by the former and latter halves. Table 5.1 shows some statistics related to this dataset. The dataset was preprocessed using the following software:

- English syntactic (HPSG) parser: Enju[3](Miyao and Tsujii, 2008)
- English tokenizer: stepp (included in Enju)
- Japanese tokenizer: Mecab[4] (with ipadic-2.7.0)

The followeing approaches were compared in the experiments:

- Baseline: baseline one-step SAMT without pre-/post-ordering
- Preorder: two-step SMT with pre-ordering (Katz-Brown and Collins, 2008) and SAMT
- Postorder: two-step SMT with the proposed method

[3]http://www.nactem.ac.uk/tsujii/enju/index.html
[4]http://mecab.sourceforge.net/

Table 5.1: Data statistics. HFE stands for Head Final English.

	Training	Dev.	Test
#sentence	3,189,025	1,000	1,000
#Japanese word	115,877,483	37,066	35,921
#English word	105,966,236	33,096	33,376
#HFE word	100,169,813	31,228	31,331

SAMT has a large advantage as regards long distance reordering because of its hierarchical approach and it also works much better (~2% or more in BLEU, TER, and RIBES) than phrase-based (PBMT) (Koehn et al., 2003) and hierarchical phrase-based (HPBMT) (Chiang, 2007) methods (The results with PBMT and HPBMT will be presented later in this section for reference.).

SAMT rule tables for Baseline and Preorder were trained using a standard Moses workflow with automatic word alignment by MGIZA++[5], `grow-diag-final` bidirectional alignment heuristics, and rule extraction considering unlimited word span. The PBMT phrase table for lexical translation in Postorder was trained with MGIZA++, `grow-diag-final-and` heuristics, and phrase extraction up to 7 words. Here, sentence pairs with a source or target side longer than 64 words were eliminated before word alignment by MGIZA++, to avoid any problematic underflow. The SAMT rule table for Postorder was trained with obvious word alignments between HFE and English and rule extraction taking account of the unlimited word span. The language models were word 5-gram language models of English and HFE, trained by SRILM[6].

The following four evaluation metrics were used in this experiments:

- BLEU (Papineni et al., 2002) by `mteval-v13a.pl`
- TER (Snover et al., 2006) by `tercom-7.25.jar`
- RIBES (Isozaki et al., 2010a) by `RIBES-1.02.3`[7]
- PER (Tillmann et al., 1997)

[5]http://sourceforge.net/projects/mgizapp/

[6]http://www-speech.sri.com/projects/srilm/

[7]http://www.kecl.ntt.co.jp/icl/lirg/ribes/

5.3.2 Results

First, the efficiency of two-step SMT with the proposed method was investigated. Two parameter values in `moses_chart` were varied to determine the relationship between accuracy and efficiency: the maximum chart span (`-max-chart-span`) and the stack size for cube pruning (`-cbp`). To realize long distance reordering, the maximum chart span was basically set to 999. With the proposed method, Japanese-to-HFE translation was a monotonic PBMT with a very small stack size (`-dl 0 -s 5` in `moses`). Table 5.2 compares decoding times with similar translation accuracies. The two-step SMT with the proposed method ran about six times faster than the baseline one-step SAMT with comparable translation accuracy. This shows the empirical efficiency of the two-step SMT with the proposed method compared with the one-step SAMT. The two-step SMT with pre-ordering was much worse than the other two methods and was not improved by employing a longer reordering limit or larger stack size.

Next, the results obtained with similar decoding time were compared. Table 5.3 shows scores obtained with a decoding time similar to that of the two-step SMT with the proposed method (1.48 seconds per sentence). The baseline one-step SAMT was much worse than that in Table 5.2, although its decoding time became as fast as that of the two-step SMT with the proposed method. This clearly shows that a standard SAMT does not work in a short decoding time due to severe search errors in its too restricted search space.

Finally, the results obtained with larger search spaces using unlimited chart spans with larger stack sizes were compared. Table 5.4 shows scores for baseline one-step SMT and two-step SMT obtained with the proposed method with larger stack sizes. The baseline achieved its best performance, which is even better than that of the proposed method, with a stack size 500 for TER, and 1000 for RIBES. This suggests that the SAMT was potentially the most accurate, although it was very slow in practice. The two-step SMT with the proposed method did not improve with larger search spaces and seemed to reach its upper bound with a stack size of 100.

Table 5.2: Decoding times (second per sentence and time ratio to Postorder) with similar translation accuracies in BLEU, TER, RIBES, and PER. L_r stands for reordering limit (`-max-chart-span`) and L_c stands for stack size in cube pruning (`-cbp`).

Method	L_r	L_c	BLEU	TER	RIBES	PER	time (sec.)
Baseline	999	50	0.307	0.625	0.733	0.398	9.48 (6.4x)
Preorder	20	100	0.290	0.641	0.714	0.394	1.43 (1.0x)
Postorder	999	100	0.311	0.625	0.734	0.393	1.48

Table 5.3: BLEU, TER, RIBES, and PER scores with similar decoding time.

Method	L_r	L_c	BLEU	TER	RIBES	PER	time (sec.)
Baseline	20	5	0.284	0.645	0.719	0.408	1.42 (1.0x)
Postorder	999	100	0.311	0.625	0.734	0.393	1.48

Table 5.4: BLEU, TER, RIBES, and PER scores with larger search space.

Method	L_r	L_c	BLEU	TER	RIBES	PER	time (sec.)
Baseline	999	100	0.311	0.622	0.736	0.396	12.2 (8.2x)
		500	0.313	0.617	0.741	0.395	17.9 (12x)
		1000	0.313	0.620	0.743	0.395	23.5 (16x)
Postorder	999	500	0.313	0.624	0.734	0.392	3.19 (2.2x)
		1000	0.312	0.624	0.735	0.392	5.19 (3.5x)

Table 5.5: Number of rules in rule tables for baseline one-step SAMT and proposed post-ordering.

Method	# rules (filtered with test set)
Baseline	25,073,976
Postorder	3,225,339

Table 5.6: Translation Examples.

(a) Example 1

Source	図5は、冷却フィン34に流れるコモンモード電流の振動成分をfft解析したものである。
Reference	fig. 5 is a view showing the fft analysis results of the vibration component of the common mode current flowing into the cooling fin 34.
Baseline	fig. 5 is a vibration component of the common mode current flowing through the cooling fins 34 fft analyzed.
HFE	fig. 5 _va0 cooling fin 34 through common mode current of vibration component of fft analysis is.
Postorder	fig. 5 is a fft analysis of the vibration component of the common mode current through the cooling fins 34.

(b) Example 2

Source	本実施形態は、本発明を限定するものではない。
Reference	the embodiments are not intended to limit the present invention.
Baseline	the present embodiment is not limited to those of the present invention.
HFE	present embodiment _va0 present invention _va1 limited not is.
Postorder	the present embodiment the present invention is not limited.

(c) Example 3

Source	ステップs11において、プライマリプーリ11への入力トルクを計算する。
Reference	in a step s11, an input torque to the primary pulley 11 is calculated.
Baseline	in step s11, the input torque to the primary pulley 11 is calculated.
HFE	s11 step in, primary pulley 11 to input torque _va2 calculates.
Postorder	in step s11, the input to the primary pulley 11 calculates the torque.

5.3.3 Discussion

Trade-off between Accuracy and Efficiency in Post-ordering Approach

There is generally a trade-off between accuracy and efficiency with the search approximation. Simple approximations by narrowing rule size, stack size and reordering limit

certainly increase efficiency but often cause severe degradation in accuracy, as shown in the baseline results in Tables 5.2 and 5.3. Table 5.5 compares the number of SAMT rules used for decoding of the test set[8] by the baseline SAMT and the proposed post-ordering. The baseline SAMT has about eight times more rules than the proposed method. The function of these rules was to handle much larger lexical translation ambiguities in the baseline SAMT but they also caused a significant increase in the decoding time. Contrary, the proposed post-ordering ran efficiently using much smaller number of rules by excluding lexical translation ambiguities. Although the proposed method limits lexical translation to 1-best hypotheses only, these hypotheses are constrained by the intermediate source-ordered target language. The constrained approximation enables us to realize a more effective lexical translation than simply narrowing the search space in the integrated search. In Table 5.6(a), the baseline one-step SAMT failed to reorder "fft" and "analysis" while the two-step SMT with the proposed method successfully moved them toward the top of the complement. The baseline tries to solve word translation and reordering jointly and so sometimes fails to search long distance reordering due to its limited search space.

The proposed approach has possible problems that may degrade its accuracy; intermediate lexical translation results may differ from the ideal source-ordered target language due to lexical translation errors. The problem was not very severe in the above experiments but may become serious especially when there are insufficient training data and when many unknown words appear. Table 5.6(b) shows an example of this kind of error. The HFE sentence in Example 2 had two pseudo-particles for subjects, _va0 and _va1. As a result, the preceding noun phrases "present embodiment" and "present invention" were both moved incorrectly toward the beginning of the sentence. This kind of error may occur owing to the ambiguity of Japanese particles in their syntactic roles. Since the baseline SAMT could translate this sentence successfully, it can be seen as a side effect of the proposed method. Example 3 presents another type of error caused by a passive construction in the source Japanese sentence. The HFE sentence should be "s11 step in, primary pulley 11 to input torque _va2 calculated is" in the passive voice. However, the active-voice verb "calculates" was used without its corresponding subject, which meant the post-ordering could not generate a syntactically correct sentence. The passive voice is very commonly used in Japanese

[8] We filtered rules to eliminate unnecessary ones for decoding the test set using `filter-rule-table.py`.

and sometimes causes serious translation errors especially with SMT. The baseline SAMT could translate it using appropriate SAMT rules, but the proposed post-ordering did not rewrite "calculates" in HFE as "is calculated". These two problems relate to shortcomings with the two-step SMT as regards post-ordering. The post-ordering depends strongly on syntactic clues in HFE such as pseudo-particles and prepositions to determine syntactic constraints for hierarchical reordering. However, if these clues are not correct, the post-ordering has to explore hypotheses satisfying those *wrong* constraints and tends to generate syntactically incorrect English sentences.

Difficulty of Isolated Lexical Translation and Post-ordering

The above experimental results suggest that separating the lexical and post-ordering problems can be achieved more easily than finding a solution to the integrated problem. We quantitatively analyzed their respective difficulties. Tables 5.7 and 5.8 show stage-wise results for Japanese-to-HFE monotone lexical translation and HFE-to-English post-ordering, respectively.

Surprisingly, as shown in Table 5.7, the use of larger stack sizes did not help to improve the lexical translation accuracy of all the evaluation metrics. This means that the lexical translation with the proposed method was unambiguous and could be solved very efficiently with a small stack size.

Table 5.8 shows the stage-wise results of the HFE-to-English post-ordering with different reordering limits, using the oracle HFE sentences taken from English reference sentences as the input. The scores were very high, especially with a large maximum chart span. This suggests that the HFE-to-English post-ordering can be achieved effectively and efficiently by SAMT.

Standard phrase-based SMT (PBMT) and hierarchical phrase-based SMT (HPBMT) were also applied to the post-ordering problem. Table 5.9 shows the results. HPBMT was slightly better in RIBES than PBMT but worse than SAMT. HPBMT also captured a hierarchical structure but had fewer constraints with respect to constituent types compared with SAMT. PBMT focused more on local context so its BLEU was better than that of HPBMT. Larger reordering limit values were also tested but showed unsuccessful results; this clearly suggests that the phrasal reordering model used in PBMT is not sufficient for

Table 5.7: Stage-wise evaluation results of Japanese-to-HFE monotone lexical translation (evaluated with oracle HFE sentences HFE_{oracle} taken from the English reference sentences). L_r stands for reordering limit (always zero in this table) and L_s stands for stack size in (`-s`).

Method	L_r	L_s	BLEU	TER	RIBES	PER	time (sec.)
Lexical Translation (Ja-to-HFE)	0	5	0.347	0.581	0.773	0.386	0.104
		20	0.347	0.581	0.773	0.386	0.162
		50	0.347	0.580	0.773	0.385	0.365
		100	0.347	0.581	0.773	0.386	0.555

Table 5.8: Stage-wise evaluation results of HFE-to-English translation (using oracle HFE sentences HFE_{oracle} as inputs). L_r stands for reordering limit (`-max-chart-span`) and L_c stands for stack size in cube pruning in (`-cbp`, always 100 in this table).

Method	L_r	L_c	BLEU	TER	RIBES	PER	time (sec.)
Postorder (HFE-to-En)	15	100	0.669	0.260	0.829	0.0750	0.471
	20		0.699	0.225	0.865	0.0703	0.568
	24		0.713	0.211	0.880	0.0676	0.771
	28		0.722	0.201	0.892	0.0677	0.809
	999		0.742	0.179	0.914	0.0673	1.35

Table 5.9: Comparison of SAMT, HPBMT, and PBMT in post-ordering. L_r stands for reordering limit (`-max-chart-span` for SAMT and HPBMT, `-distortion-limit` for PBMT) and L_c stands for stack size (`-cbp` for SAMT and HPBMT, `-s` for PBMT).

Method	L_r	L_c	BLEU	TER	RIBES	PER	time (sec.)
SAMT	999	100	0.311	0.625	0.734	0.393	1.48
HPBMT	999	100	0.288	0.646	0.713	0.397	1.23
PBMT	16	100	0.299	0.646	0.699	0.392	1.21

long distance reordering in Japanese-to-English translation.

5.4 Related Work

Reordering is a theoretically and practically challenging problem in SMT. In early SMT studies, reordering was modeled by distance-based constraints in the translation model (Brown et al., 1993; Koehn et al., 2003). This reordering model is easy to compute and also works with relatively similar language pairs such as French-to-English. Recent PBMT studies have employed lexicalized phrasal reordering models (Tillmann, 2004; Nagata et al., 2006; Galley and Manning, 2008) to constrain phrasal orientation using lexical information. These models do not directly model long distance reordering and are insufficient for Japanese-to-English SMT. On the other hand, the use of syntax in SMT (Yamada and Knight, 2001; Galley et al., 2004; Graehl and Knight, 2004; Zollmann and Venugopal, 2006) are theoretically sound solutions for the reordering problem based on syntactic constraints. Hierarchical phrase-based MT (Chiang, 2007) employed a formally syntactic structure between the source and target languages (Wu, 1997). Treelet translation (Quirk et al., 2005) employed an isolated subtree reordering models. Although these tree-based models can achieve long distance reordering in their hierarchical representations, their search involves large computational complexity.

A novel approach to reordering, called pre-ordering, has been studied in recent years. Syntax-based methods have been applied to various language pairs (Xia and McCord, 2004; Collins et al., 2005; Li et al., 2007; Xu et al., 2009; Hong et al., 2009; Genzel, 2010; Katz-Brown et al., 2011). These syntax-based methods were recently extended by automatically induced parsers (DeNero and Uszkoreit, 2011; Neubig et al., 2012).

The post-ordering framework also relates to post-editing technologies, which aim to correct errors in a rule-based translation (Simard et al., 2007; Dugast et al., 2007; Ehara, 2007) or a different type of SMT (Aikawa and Ruopp, 2009). There is a major difference between post-ordering and post-editing; in the post-editing framework, the preceding translation process is a complete source-to-target translation, and post-editing itself mainly provides error correction. In contrast, the SMT framework with post-ordering divides the entire translation problem into translation and reordering subproblems. It has an advantage in that the subproblems can be easily and efficiently solved compared with the post-editing

approach involved in a complete translation process[9]. Pivot translation (Wu and Wang, 2007) is also similar, in that it solves two translation problems sequentially. The largest difference between pivot translation and the post-ordering approach is their intermediate language; the pivot translation uses another language (typically English) while the post-ordering approach uses reordered target language.

5.5 Conclusion

This work presented a novel syntax-based post-ordering method for efficient Japanese-to-English SMT with long distance reordering, using Japanese-ordered English induced by a reordering method for English-to-Japanese. The proposed method provides a practical alternative to syntax-based SMT by approximately decomposing it into monotone lexical translation and syntax-based post-ordering with the intermediate language, HFE. It empirically provides a six-fold reduction in the decoding time with a comparable translation accuracy; it could decode a sentence more than six times faster than a standard SAMT in a Japanese-to-English patent translation.

This post-ordering is an extension of the reordering model presented by (Bangalore and Riccardi, 2000) and also can be integrated with lexical translation as in common phrase-based SMT methods. This increases the time complexity to some extent but is also expected to balance accuracy and efficiency for long distance reordering.

[9]The implementation in this work does not exclusively isolate lexical translation and post-ordering; some degree of word translation is also allowed in the post-ordering to recover modified and eliminated words in the intermediate language (described in section 5.2.2).

Chapter 6

Japanese-to-English Translation System for Patents

A Japanese-to-English patent SMT system is developed integrating the proposed techniques: the patent-adapted Japanese word segmentation (Chapter 3), the unknown katakana word transliteration bootstrapped from a parallel corpus (Chapter 4), and the syntax-based post-ordering (Chapter 5).

An advantage of the post-ordering framework is that it is easy to integrate the domain-adapted word segmentation and the unknown word transliteration in its lexical translation step and that the following reordering step can use the improved lexical translation results. To implement the same thing in the pre-ordering as Hoshino et al. (2013b), the system needs domain adaptation of a Japanese syntactic parser in addition to the word segmenter, and a tight integration of transliteration into the SMT decoding.

The syntactic parsing plays a very important role in the syntax-based SMT, but most Japanese syntactic parser do not work well in the patent domain due to characteristics of patents different from general domains (typically newspapers used in parsing studies). In contrast, English syntactic parsing have been studied on technical documents such as biomedical articles (Miyao et al., 2008). Such a parser (e.g., Enju (Miyao and Tsujii, 2008)) also works relatively well in the patent domain compared to ones trained using newspaper data (Isozaki et al., 2012). Since domain adaptation of syntactic parsers is much more difficult than that of word segmenters, the post-ordering SMT with such a English parser is

practically useful.

The transliteration step usually implemented as a postprocessing of the SMT to transliterate untranslated words, but it gives no effect on reordering in a standard one-pass SMT. Durrani et al. (2014) proposed a back-off transliteration method to address this problem, but it requires a special treatment of such a back-off model in the SMT decoder. The proposed system use the transliteration as a postprocessing of the first-pass lexical translation and the resulting transliterations are used in the second-pass syntax-based reordering. This enables a simple and straightforward integration of the transliteration in the SMT.

This chapter presents the proposed system with its evaluation results. The results showed this integrated system further improves the Japanese-to-English patent SMT with the post-ordering in Chapter 5, by its better lexical translation derived from patent-adapted word segmentation and term transliteration.

6.1 System Architecture

The system is based on large-scale language resources in the patent domain, a Japanese-English parallel corpus and monolingual corpora of Japanese and English. The workflow of the SMT system is illustrated in Figure 6.1. The translation is divided into the following four processes by the techniques proposed in this thesis work.

1. Japanese word segmentation using a patent-adapted word segmentation model
2. Translation into an intermediate language, Head Final English (HFE), by a monotone phrase-based SMT
3. Transliteration of untranslated Japanese katakana words (i.e. unknown words in the previous process) into English words, by a monotone phrase-based SMT in the character level
4. Post-ordering into English by a syntax-based SMT

The models are trained as follows:

- The word segmentation model is trained using a general domain labeled (word segmented) corpus and a patent unlabeled (not word segmented) corpus as described in Chapter 3.

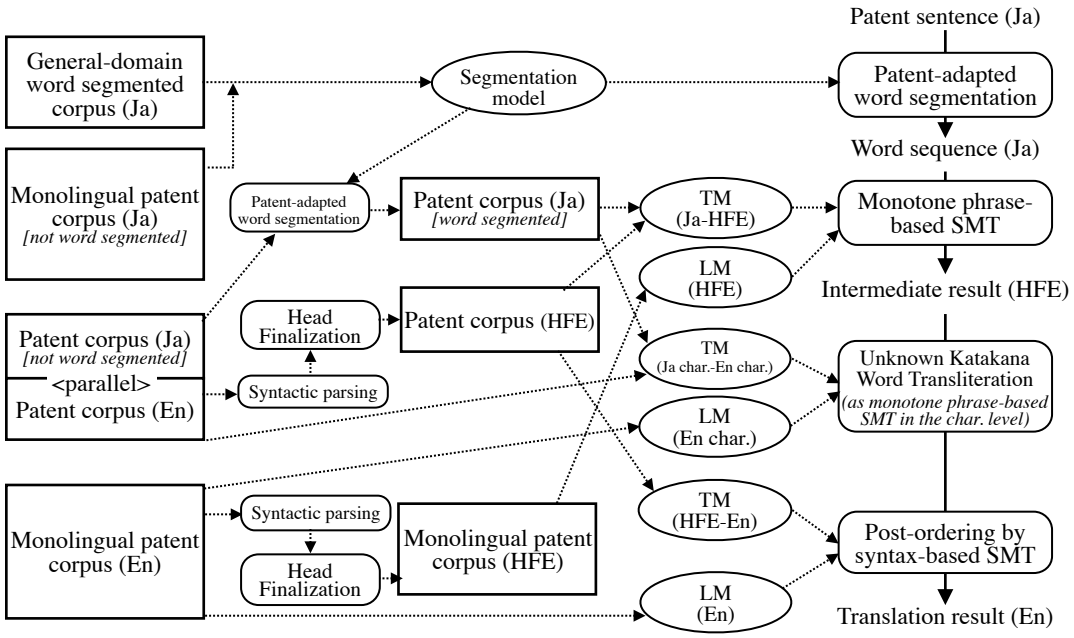


Figure 6.1: Training and translation workflow by the patent-oriented Japanese-to-English SMT.

- The transliteration model is trained using transliteration pairs mined from the parallel corpus as described in Chapter 4.

- The translation models used in the monotone phrase-based SMT and syntax-based SMT are trained using the parallel corpora as described in Chapter 5. Since HFE can be generated automatically by the Head Finalization rules, we can easily obtain the parallel corpus of three languages: Japanese, English, and HFE.

- The language models are trained using the monolingual corpora. The monolingual HFE corpus is generated similarly to the HFE portion of the parallel corpus.

The proposed system has a bit complex architecture compared to a standard SMT system composed of an off-the-shelf word segmenter and a one-pass SMT, but it tackles common practical problems in the patent SMT discussed in this thesis.

Tokenizer	Training (2,862,022 sents.)	Development (2,000 sents.)	Test9 (2,000 sents.)	Test10 (2,300 sents.)
Proposed	95,465,533	75,020	75,962	101,309
Baseline	94,914,460	74,627	75,504	100,589
KyTea	101,718,532	80.025	80,842	107,405
MeCab	93,030,977	73,263	74,066	99,163
JUMAN	91,052,206	71,707	72,515	97,205
English	88,192,234	68,854	69,806	94,906

Table 6.1: Bilingual corpus statistics in the number of words for translation experiments.

6.2 Implementation

Here the implementation of the proposed system is summarized with respect to language resources and software components.

6.2.1 Language Resources

Japanese-to-English patent translation dataset used in NTCIR-9 Goto et al. (2011) and NTCIR-10 Goto et al. (2013) PatentMT were used for the system. The NTCIR-9 and NTCIR-10 datasets shared the same training and development sets and used different test sets. Its bilingual corpus statistics are shown in Table 6.1. Its monolingual corpora in Japanese (540 million sentences) and English (370 million sentences, 11 billion words) were also used.

The English sentences were tokenized and parsed by an English syntactic parser Enju with its "GENIA" models for biomedical articles, and then lowercased. The HFE sentences were obtained from the English sentences by Head Finalization rules. The Japanese sentences were tokenized by the proposed patent-adapted word segmenter. Several different word segmenters were also compared: the baseline word segmenter, and three public available ones (KyTea, MeCab, and JUMAN), for comparison with the proposed one. Here in the training set, long sentences exceeding 64 words in either Japanese or English were filtered out. The segmentation results by KyTea were used for the long sentence filtering

because it is based on a short word unit and resulted in the largest number of segmented words. Note that the sentence set was the same for all Japanese segmenters.

6.2.2 Components

Japanese Word Segmentation

The system uses the patent-adapted word segmenter that worked best in the experiments in section 3.3, the CRF-based segmenter with the BE features from 550 million sentences and the PD features from 10 million sentences. The segmenter uses a CRF implementation CRFsuite[1], and the n-gram-based BE values are stored in an efficient data structure of KenLM[2] for fast feature extraction from Japanese sentences to be segmented.

Japanese-to-HFE Monotone Phrase-based MT

The Japanese-to-HFE monotone PBMT is implemented with Moses (version 2.1), which is a newer version than that used in the experiments in section 5.3. Its phrase table was trained using the Japanese-HFE parallel sentences with MGIZA++ word alignment and grow-diag-final-and alignment symmetrization heuristics, limiting the maximum phrase length to seven. The reordering limit is set to zero, but a standard lexicalized reordering model (`wbe-msd-bidirectional-fe`) is used to constrain adjacent phrase translations. The language model is a word 6-gram language model with interpolated modified Kneser-Ney smoothing trained with KenLM (with the option "`-prune 0 0 1`" to prune singletons for orders three and higher) using the HFE monolingual corpus. The model weights were optimized in BLEU Papineni et al. (2002) using Minimum Error Rate Training (MERT) Och (2003). The best weights were chosen among ten individual runs of MERT.

Katakana Transliteration

The transliteration model is a Moses-based monotone PBMT in the character level. Its character-based phrase table was trained using the set of extracted transliteration fragments from the katakana-English phrases used in the experiments in section 4.3 (Proposed

[1] http://www.chokkan.org/software/crfsuite/

[2] https://kheafield.com/code/kenlm/

with FullyAligned), with MGIZA++-based re-alignment and grow-diag-final-and alignment symmetrization heuristics, limiting the maximum phrase length to seven. The reordering limit is set to zero, and no reordering model is used. The language model is a character 9-gram with interpolated modified Kneser-Ney smoothing trained with SRILM using the English character sequences from the English monolingual corpus. The model weights were optimized in BLEU in the character level using MERT.

HFE-to-English Syntax-based MT

The HFE-to-English syntax-based was implemented with Moses-chart and trained using the HFE sentences and the corresponding English parse trees. Its reordering parameter `max-chart-span` was set to 200 to allow arbitrary distance reordering for accurate Japanse-to-English translation[3]. The search space parameter `cube-pruning-pop-limit` was set to 32 for efficiency. The language model is a word 6-gram one trained similarly to that for the preceding PBMT, using the English monolingual corpus. The model weights were the best ones among ten individual runs of MERT to optimize BLEU.

6.3 Evaluation

The performance of the proposed system by the following experiments was evaluated. The experiments were basically similar to the ones in Chapter 5, but used different, latest NTCIR test sets. The main concern in the experiments were effects of the domain-adapted word segmentation and unknown word transliteration on the post-ordering SMT in the proposed system. Several SMT configurations were compared for the evaluation: with different word segmenters, with and without transliteration.

6.3.1 Compared Methods

The following segmenters were compared for the translation experiments.

- Baseline: the baseline word segmenter trained only using the labeled data and the baseline features

[3]It exceeded the maximum sentence length in the development and test sets.

- Proposed: the patent-adapted segmenter using the labeled general-domain corpus and the large-scale unlabeled patent corpus with the BE and PD features
- KyTea, MeCab, and JUMAN[4]: publicly available Japanese morphological analyzers

The results by the post-ordering were also compared with those by standard SAMT and PBMT. The search space parameters of the standard SAMT were set to the same value as the HFE-to-English SAMT, to compare the performance with similar computation time[5].

6.3.2 Results and Discussion

Table 6.2 shows the translation performance in BLEU and TER with the results of statistical significance tests (p=0.05) by bootstrap resampling (Koehn, 2004), in which the overall system resulted in the best. The table also shows the results of intermediate Japanese-to-HFE translation. The advantage of the system can be attributed to three techniques included in the system: domain adaption of word segmentation, katakana unknown word transliteration, and post-ordering.

First, the post-ordering contributed the largest and significant improvements compared with the standard SAMT and PBMT, by about 1-2 points in BLEU and 2-3 points in TER. They basically followed the results by (Sudoh et al., 2013d).

Second, the proposed word segmentation showed significant improvements in most cases, by the better intermediate translation results shown at the bottom of Table 6.2. Although the absolute improvement was not so large, the domain adaptation worked consistently. These results suggest that the domain adaptation of word segmentation actually worked for the patent SMT. The advantage of the patent-adapted word segmentation was also analyzed by the number of unknown words in translation. Table 6.3 shows the numbers of unknown kanji and katakana words that were not translated in the monotone PBMT, by the five word segmenters in the experiments. These values reflect the consistency and granularity problem in word segmentation (Chang et al., 2008). If the word segmentation

[4]http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN

[5]Actually the post-ordering needs the time for the first monotone PBMT but it ran very fast and did not affect so much Sudoh et al. (2013d).

System	Ja word segmenter	Test9		Test10	
		BLEU (%)	TER (%)	BLEU (%)	TER (%)
Overall system (Ja-HFE monotone PBMT + transliteration + HFE-En SAMT)	Proposed	**34.77**	$^{+}$**51.86**	**35.75**	**50.71**
	Baseline	$^{+}$*34.29	$^{+}$*52.16	$^{+}$*35.21	$^{+}$50.90
	KyTea	$^{+}$*34.42	$^{+}$*52.30	*35.37	*51.38
	MeCab	$^{+}$*34.52	$^{+}$*52.21	$^{+}$*35.41	$^{+}$*50.92
	JUMAN	$^{+}$34.59	$^{+}$52.10	$^{+}$*35.41	$^{+}$*51.00
Post-ordering (Ja-HFE monotone PBMT + HFE-En SAMT)	Proposed	**34.75**	**51.90**	**35.71**	**50.71**
	Baseline	*34.18	*52.30	*35.14	50.96
	KyTea	*34.32	*52.40	*35.33	*51.41
	MeCab	*34.33	*52.35	*35.28	*51.03
	JUMAN	*34.50	52.19	*35.35	*51.07
SAMT (efficiency-oriented)	MeCab	*33.11	*53.26	*33.67	*52.19
PBMT (distortion limit=12)	MeCab	*31.96	*55.04	*33.06	*53.77
Ja-HFE monotone PBMT	Proposed	**35.87**	**49.05**	**37.11**	**48.19**
	Baseline	*35.30	*49.46	*36.46	*48.58
	KyTea	*35.50	*49.77	*36.66	*48.86
	MeCab	*35.45	*49.57	*36.71	*48.54
	JUMAN	35.70	*49.42	*36.65	*48.73

Table 6.2: Results of overall Japanese-to-English translation and intermediate Japanese-to-HFE translation in BLEU and TER. $^{+}$ indicates the difference from the results without transliteration is statistically significant. * indicates the difference from Proposed in the same group is statistically significant.

is consistent and have relatively small granularity (choosing shorter words), the number of the unknown words becomes small. The granularity is closely related to the problem of compound words in this work; the translation of compound words becomes easy if they are segmented to short and appropriate component words. MeCab and JUMAN are dictionary-based word segmenters that have an advantage on precise segmentation of in-vocabulary words. JUMAN used a large-scale dictionary collected from web texts covering many

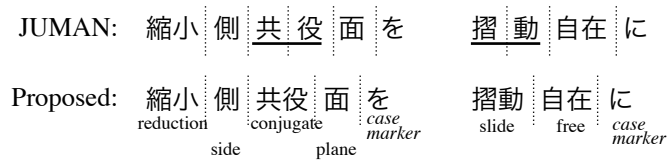


Figure 6.2: Examples of small granularity segmentation for out-of-vocabulary words by JUMAN.

domain-specific words, and resulted in a smaller number of unknown words than MeCab. KyTea and this paper's segmenter are character-based ones that have an advantage on identifying out-of-vocabulary words (as shown in Table 3.2). KyTea worked well on kanji words, but derived a large number of katakana unknown words. It was probably due to the difference of embedded information between ideogram (kanji) and phonogram (katakana). Katakana compound words are usually difficult to segment only by their poor character-based information. The proposed method used reliable word boundary clues derived from the large-scale corpora and achieved consistent word segmentation of katakana compound words with more appropriate granularity than others, as suggested by the smallest number of katakana unknown words in Table 6.3. Such an advantage was not found in kanji words compared to KyTea and JUMAN. However, JUMAN tended to choose small granularity segmentations for out-of-vocabulary words as shown in the examples in Figure 6.2, so these results may not indicate directly the disadvantage of the proposed method.

Finally, the transliteration itself did not improve BLEU and TER significantly in the system, although some significant improvements were found in the results by the other segmenters because of their many unknown katakana words. Its effect was limited only on the unknown katakana words and their context words (related to the word n-gram language model and the post-ordering) and did not contribute well to BLEU and TER with a small number of the unknown katakana words. The transliteration accuracy in the intermediate HFE results with the transliteration was analyzed as shown in Table 6.4. About a half of the unknown katakana words were transliterated correctly. This improvement is practically important for the assimilation.

Ja word segmenter	test9		test10	
	kanji	katakana	kanji	katakana
Proposed	18 (18)	30 (20)	29 (23)	34 (20)
Baseline	54 (43)	87 (59)	98 (71)	101 (59)
KyTea	10 (10)	108 (79)	14 (14)	132 (78)
MeCab	48 (39)	68 (50)	100 (73)	87 (55)
JUMAN	2 (2)	48 (41)	9 (9)	71 (45)

Table 6.3: Statistics of unknown kanji and katakana words (non-translated words by monotone PBMT). The numbers in parentheses are the number of unique unknown words.

Test9	Test10
53.33 (16/30)	59.37 (19/32)

Table 6.4: Transliteration accuracy in sample-wise correctness (ACC) in the proposed system.

6.4 Conclusion

This chapter presented our Japanese-to-English SMT system specialized for patent translation, including the effective word segmentation by our domain adaptation method, the unknown katakana word transliteration, and the efficient syntax-based post-ordering. The system achieved better translation performance than those using existing Japanese word segmenters and standard SMT methods.

Chapter 7

Conclusions

This thesis addressed Japanese-to-English SMT for technical documents such as patents. The target MT task is beneficial for practical industrial needs such as surveys and distribution of technical information written in Japanese. The translation of such technical documents can be literal and suitable for the MT but has problems on technical terms and long sentences, which are distinguished ones from other translation tasks in different language pairs and domains. This thesis work focused on these problems in this target task: Japanese word segmentation, unknown word translation, and long distance reordering.

The first problem was the word segmentation of uncommon technical terms in Japanese patents. This thesis work proposed the use of the branching entropy as word segmentation clues in discriminative semi-supervised Japanese word segmentation with very large-scale unlabeled Japanese patent corpora for adapting word segmentation to the patent domain. It works better than the existing methods using the accessor variety, by probabilistic characteristics of the BE independent from the corpus size. This enables effective Japanese word segmentation with no additional human annotations using unlabeled patent corpora together with the limited number of existing general-domain labeled corpora.

The second problem was unknown technical terms that cannot be translated due to the lack of bilingual correspondence in the SMT training data. This thesis work focused on the fact that more than a half of these unknown words are transliterated katakana words. Statistical transliteration was used as a character-based machine translation using the transliteration fragments extracted from the sentence-aligned bilingual patent corpora. This the-

sis work proposed a novel noise-aware alignment method that can identify partial noise in transliteration candidates, and improved transliteration accuracy for unknown katakana words in the patent dataset.

The third problem was the long distance reordering in long patent sentences that causes very large computational complexity in the Japanese-to-English SMT. This thesis work proposed a novel SMT framework called post-ordering, as opposed to the pre-ordering that is very effective in English-to-Japanese direction. The proposed post-ordering conducts lexical translation firstly and then conducts reordering based on a syntax-based SMT technique, to obtain syntactically motivated translation results in English. Although the post-ordering is an approximation of the general syntax-based SMT, it can reduce the computational cost largely without performance drop in the translation accuracy.

Finally a patent SMT system was developed using these techniques within the Moses-based SMT framework. The system achieved the better translation evaluation scores than several baselines with general-purpose word segmenters and the standard SMT techniques.

7.1 Contributions of this Thesis Work

A main contribution of this thesis work is the development of practical techniques for the Japanese-to-English SMT for technical documents. Most of previous MT and NLP studies were established on general domain data such as newspaper articles. This thesis work focused on MT problems on technical documents that involved different essential problems from other types of documents. Although this thesis work used patent data as the major target, the approaches can also be applied to other kinds of technical documents with large-scale document archives such as manuals and research articles. Since translation of such specialized documents usually requires expert knowledge for human translators, the proposed techniques are beneficial to practical MT with less human efforts.

With respect to the individual problems, the contribution of this thesis work is two-fold. First, most previous studies on Japanese-to-English patent SMT did not address the technical term problem but applied existing general domain word segmenters without transliterating unknown words. This thesis work demonstrates the use of monolingual and bilingual patent corpora actually helps overcome the unknown word problem without additional hu-

man annotations. Second, the long-distance reordering is one of the most important problem in the field of SMT. In contrast to the previous studies that use the pre-ordering based on the source language syntax, this thesis work proposed a novel post-ordering framework based on the target language syntax. This increased efficiency of the syntax-based MT with the target language syntax that was computationally more expensive than that with the source language syntax.

7.2 Future Work

There are some future prospects of further studies on SMT from this thesis work.

First, it is important to expand translation target domains of SMT. Domain adaptation for resource-poor domains is a promising direction. There may be no sufficient document archives nor bilingual documents in demanded domains. Although some studies tried to realize domain adaptation by a model mixture, that is not sufficient for translating domain-specific terms. Acquiring a domain-specific lexicon from non-parallel bilingual language resources is very important for such a purpose.

Second, there are different translations that have to be distinguished according to contexts and domains. For example in patents, a word may have different standard terms in the target language, or different meanings in different technical fields. This kind of term consistency or ambiguity cannot be handled by the standard SMT framework and causes serious misunderstanding. Context-awareness related to document and discourse structure, and coreference resolution is a more challenging problem. These problems motivates document-wise MT towards high-quality MT just like expert human translators.

Finally, we need to explore better automatic evaluation methodologies reflecting actual understandability. Since the current evaluation metrics only focuses on unweighted word-based agreement, they cannot penalize serious translation errors, for example a negative sentence is translated into an affirmative sentence by dropping a negation expression. This kind of problem must be handled also in the SMT modeling for more practical MT.

Bibliography

Takako Aikawa and Achim Ruopp. 2009. Chained System: A Linear Combination of Different Types of Statistical Machine Translation Systems. In *Proceedings of 12th Machine Translation Summit*.

Yaser Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 400–408, July.

Rie Kubota Ando and Lillian Lee. 2003. Mostly-unsupervised statistical segmentation of Japanese kanji sequences. *Natural Language Engineering*, 9(2):127–149.

Srinivas Bangalore and Giuseppe Riccardi. 2000. Finite-State Models for Lexical Reordering in Spoken Language Translation. In *Proceedings of the Sixth International Conference on Spoken Language Processing*, pages 422–425.

Srinivas Bangalore, Patrick Haffner, and Stephan Kanthak. 2007. Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 152–159, Prague, Czech Republic, June.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Jennifer C. Lai, and Robert L. Mercer. 1992. An Estimate of an Upper Bound for the Entropy of English. *Computational Linguistics*, 18(1):31–40, March.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Columbus, Ohio, June.

Stanley F. Chen and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Harvard University.

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

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540, Ann Arbor, Michigan, June.

Marta R. Costa-jussà and José A. R. Fonollosa. 2006. Statistical Machine Reordering. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 70–76, Sydney, Australia, July.

John DeNero and Jakob Uszkoreit. 2011. Inducing Sentence Structure from Parallel Corpora for Reordering. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Edinburgh, Scotland, UK., July.

Loïc Dugast, Jean Senellart, and Philipp Koehn. 2007. Statistical Post-Editing on SYSTRAN's Rule-Based Translation System. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223, Prague, Czech Republic, June.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 148–153, Gothenburg, Sweden, April.

Terumasa Ehara. 2007. Rule Based Machine Translation Combined with Statistical Post Editor for Japanese to English Patent Translation. In *Proceedings of MT Summit XI Workshop on Patent Translation*.

Jason Eisner. 2003. Learning Non-Isomorphic Tree Mappings for Machine Translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 205–208, Sapporo, Japan, July.

Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor Variety Criteria for Chinese Word Extraction. *Computational Linguistics*, 30(1):75–93.

Andrew Finch and Eiichiro Sumita. 2010. A Bayesian Model of Bilingual Segmentation for Transliteration. In *Proceedings of the Seventh International Workshop on Spoken Language Translation*, pages 259–266, Paris, France, December.

Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, and Takehito Utsuro. 2008. Overview of the Patent Translation Task at the NTCIR-7 Workshop. In *Proceedings of the NTCIR-7 Workshop Meeting*, pages 389–400, Tokyo, Japan, December.

Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro, Terumasa Ehara, Hiroshi Echizen-ya, and Sayori Shimohata. 2010. Overview of the Patent Translation Task at the NTCIR-8 Workshop. In *Proceedings of the NTCIR-8 Workshop Meeting*, pages 371–376, Tokyo, Japan, June.

Takaaki Fukunishi, Andrew Finch, Seiichi Yamamoto, and Eiichiro Sumita. 2013. A Bayesian Alignment Approach to Transliteration Mining. *ACM Transactions on Asian Language Information Processing*, 12(3):no. 9, August.

Michel Galley and Christopher D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 273–280, Boston, Massachusetts, USA, May.

Dmitriy Genzel. 2010. Automatically Learning Source-side Reordering Rules for Large Scale Machine Translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 376–384, Beijing, China, August.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, July.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop. In *Proceedings of the 9th NTCIR Workshop Meeting*, Tokyo, Japan, December.

Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2012. Post-ordering by Parsing for Japanese-English Statistical Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316, Jeju Island, Korea, July.

Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K. Tsou. 2013. Overview of the Patent Machine Translation Task at the NTCIR-10 Workshop. In *Proceedings of the 10th NTCIR Conference*, Tokyo, Japan, June.

Jonathan Graehl and Kevin Knight. 2004. Training Tree Transducers. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 105–112, Boston, Massachusetts, USA, May.

Zhen Guo, Yujie Zhang, Chen Su, and Jinan Xu. 2012. Exploration of N-gram Features for the Domain Adaptation of Chinese Word Segmentation. In *Proceedings of the 1st CCF Conference on Natural Language Processing & Chinese Computing*, pages 121–131.

Gumwon Hong, Seung-Wook Lee, and Hae-Chang Rim. 2009. Bridging Morpho-Syntactic Gap between Source and Target Sentences for English-Korean Statistical Machine Translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (Short Papers)*, pages 233–236, Suntec, Singapore, August.

Ohnmar Htun, Andrew Finch, Eiichiro Sumita, and Yoshiki Mikami. 2012. Improving Transliteration Mining by Integrating Expert Knowledge with Statistical Approaches. *International Journal of Computer Applications*, 58(17):12–22, November.

Zhihui Jin and Kumiko Tanaka-Ishii. 2006. Unsupervised Segmentation of Chinese Text by Use of Branching Entropy. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Poster Sessions)*, pages 428–435, Sydney, Australia, July.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2011. Machine transliteration survey. *ACM Computing Surveys*, 43(3), April.

Jason Katz-Brown and Michael Collins. 2008. Syntactic Reordering in Preprocessing for Japanese-English Translation: MIT System Description for NTCIR-7 Patent Translation Task . In *Proceedings of the 7th NTCIR Workshop Meeting*, pages 409–414, Tokyo, Japan, December.

Jason Katz-Brown, Slav Petrov, Ryan McDonald, Franz Och, David Talbot, Hiroshi Ichikawa, Masakazu Seno, and Hideto Kazawa. 2011. Training a Parser for Machine Translation Reordering. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 183–192, Edinburgh, Scotland, UK., July.

André Kempe. 1999. Experiments in Unsupervised Entropy-Based Corpus Segmentation. In *Proceedings of Computational Natural Language Learning*, pages 7–13, Bergen, Norway, June.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Alberta, Canada, April.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*, Pittsburgh, USA, October.

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

Philipp Koehn. 2004. Statistical Significance Test for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, July.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Keigo Kubo, Hiromichi Kawanami, Hiroshi Saruwatari, and Kiyohiro Shikano. 2011. Unconstrained many-to-many alignment for automatic pronunciation annotation. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA 2011)*.

A Kumaran, Mitesh M. Khapra, and Haizhou Li. 2010. Report of news 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 21–28, July.

Wai Lam, Ruizhang Huang, and Pik-Shan Cheung. 2004. Learning phonetic similarity for matching named entity translations and mining new translations. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 286–296.

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. 2007. A Probabilistic Approach to Syntax-based Reordering for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 720–727, Prague, Czech Republic, June.

Haizhou Li, A Kumaran, Min Zhang, and Vladimir Pervouchine. 2010. Report of news 2010 transliteration generation shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 1–11, July.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July.

Kikuo Maekawa. 2007. Design of a Balanced Corpus of Contemporary Written Japanese. In *Proceedings of Symposium on Large-Scale Knowledge Resources*, pages 55–58.

Evgeny Matusov, Stephan Kanthak, and Hermann Ney. 2005. On the Integration of Speech Recognition and Statistical Machine Translation. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech - Eurospeech)*, pages 3177–3180.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature Forest Models for Probabilistic HPSG Parsing. *Computational Linguistics*, 34(1):35–80.

Yusuke Miyao, Rune Sætre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Task-oriented Evaluation of Syntactic Parsers and Their Representations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 46–54, Columbus, Ohio, June.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 100–108, Suntec, Singapore, August.

Hwidong Na, Jin-Ji Li, Jungi Kim, and Jong-Hyeok Lee. 2009. Improving Fluency by Reordering Target Constituents Using MST Parser in English-to-Japanese Phrase-based SMT. In *Proceedings of the 12th Machine Translation Summit (MT Summit XII)*, pages 276–283.

Masaaki Nagata, Kuniko Saito, Kazuhide Yamamoto, and Kazuteru Ohashi. 2006. A Clustered Global Phrase Reordering Model for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 713–720, Sydney, Australia, July.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA, June.

Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a Discriminative Parser to Optimize Machine Translation Reordering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 843–853, Jeju Island, Korea, July.

Franz Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, July.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proeedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, June.

Franz Josef Och, Nicola Ueffing, and Hermann Ney. 2001. An Efficient A* Search Algorithm for Statistical Machine Translation. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*, pages 55–62, Toulouse, France, July.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese Segmentation and New Word Detection using Conditional Random Fields. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 562–568, Geneva, Switzerland, August.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 271–279, Ann Arbor, Michigan, June.

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 469–477, July.

Steven L. Scott. 2002. Bayesian Methods for Hidden Markov Models: Recursive Computing in the 21st Century. *Journal of the American Statistical Association*, 97(457): 337–351, March.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical Phrase-Based Post-Editing. In *Proceedings of the Human Language Technologies: the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 508–515, Rochester, New York, April.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proc. AMTA*, pages 223–231.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2013. Syntax-Based Post-Ordering for Efficient Japanese-to-English Translation. *ACM Transactions on Asian Language Information Processing*, 12(3), August.

Weiwei Sun and Jia Xu. 2011. Enhancing Chinese Word Segmentation Using Unlabeled Data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 970–979, Edinburgh, Scotland, UK., July.

Weiwei Sun. 2010. Word-based and Character-based Word Segmentation Models: Comparison and Combination. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010): Posters*, pages 1211–1219, Beijing, China, August.

Christoph Tillmann, Stephen Vogel, Hermann Ney, A. Zubiaga, and H. Sawaf. 1997. Accelerated DP based Search for Statistical Translation. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, volume 5, pages 2667–2670, Rhodes, Greece.

Christoph Tillmann. 2004. A Unigram Orientation Model for Statistical Machine Translation. In *Proceedings of the Human Language Technologies: the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–104, Boston, Massachusetts, USA, May.

Roy Tromble and Jason Eisner. 2009. Learning Linear Ordering Problems for Better Translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore, August.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A Conditional Random Field Word Segmenter for Sighan Bakeoff 2005. In *Proceedings of the fourth SIGHAN Workshop on Chinese Language Processing*, pages 168–171.

Yuta Tsuboi, Hisashi Kashima, Shinsuke Mori, Hiroki Oda, and Yuji Matsumoto. 2008. Training Conditional Random Fields Using Incomplete Annotations. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 897–904, Manchester, UK, August.

Masao Utiyama and Hitoshi Isahara. 2007. A Japanese-English Patent Parallel Corpus. In *Proceedings of the 11th Machine Translation Summit*, pages 475–482.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *Proceedings of the 16th International Conference on Computational Linguistics*, volume 2, pages 836–841.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving Chinese Word Segmentation and POS Tagging with Semi-supervised Methods Using Large Auto-Analyzed Data. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 309–317, Chiang Mai, Thailand, November.

Mengqiu Wang, Rob Voigt, and Christopher D. Manning. 2014. Two Knives Cut Better Than One: Chinese Word Segmentation with Dual Decomposition. In *Proceedings of*

the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 193–198, Baltimore, Maryland, June.

Hua Wu and Haifeng Wang. 2007. Pivot Language Approach for Phrase-Based Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 856–863, Prague, Czech Republic, June. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404.

Fei Xia and Michael McCord. 2004. Improving a Statistical MT System with Automatically Learned Rewrite Patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 508–514, Geneva, Switzerland, August.

Jia Xu, Jianfeng Gao, Kristina Toutanova, and Hermann Ney. 2008. Bayesian Semi-Supervised Chinese Word Segmentation for Statistical Machine Translation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1017–1024, August.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a Dependency Parser to Improve SMT for Subject-Object-Verb Languages. In *Proceedings of Human Language Technologies: the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253, Boulder, Colorado, June.

Kenji Yamada and Kevin Knight. 2001. A Syntax-based Statistical Translation Model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France, July.

Kenji Yamada and Kevin Knight. 2002. A Decoder for Syntax-based Statistical MT. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 303–310, Philadelphia, Pennsylvania, USA, July.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006. Effective Tag Set Selection in Chinese Word Segmentation via Conditional Random Field Modeling. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, pages 87–94.

Valentin Zhikov, Hiroya Takamura, and Manabu Okumura. 2010. An Efficient Algorithm for Unsupervised Word Segmentation with Branching Entropy and MDL. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 832–842, Cambridge, MA, October.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June.

Authored Works

Chapters 3 and 6

Refereed

Katsuhito Sudoh, Masaaki Nagata, Shinsuke Mori, and Tatsuya Kawahara. 2014. Japanese-to-English Patent Translation System based on Domain-adapted Word Segmentation and Post-ordering. In *Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas*, pages 234–248, October.

Chapter 4

Refereed

Katsuhito Sudoh, Shinsuke Mori, and Masaaki Nagata. 2014. Noise-aware Character Alignment for Extracting Transliteration Fragments. *Journal of Natural Language Processing*, in press.

Katsuhito Sudoh, Shinsuke Mori, and Masaaki Nagata. 2013. Noise-Aware Character Alignment for Bootstrapping Statistical Machine Transliteration from Bilingual Corpora. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 204–209, Seattle, Washington, USA, October.

Chapter 5

Refereed

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2013. Syntax-Based Post-Ordering for Efficient Japanese-to-English Translation. *ACM Transactions on Asian Language Information Processing*, 12(3), August.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Post-ordering in Statistical Machine Translation. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 316–323, Xiamen, China, September.

Other Works

Refereed

Katsuhito Sudoh and Mikio Nakano. 2003. Post-dialogue Recognition Confidence Scoring for Improving Statistical Language Models using Untranscribed Dialogue Data. In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2003)*, pages 447–452, December.

Katsuhito Sudoh and Mikio Nakano. 2005. Post-Dialogue Confidence Scoring for Unsupervised Statistical Language Model Training. *Speech Communication*, 45(4):387–400.

Katsuhito Sudoh and Hajime Tsukada. 2005. Tightly Integrated Spoken Language Understanding using Word-to-Concept Translation. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech - Eurospeech)*, pages 429–432, September.

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. 2006. Discriminative Named Entity Recognition of Speech Data Using Speech Recognition Confidence. In *Proceedings of the Ninth International Conference on Spoken Language Processing*, pages 337–340, September.

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. 2006. Incorporating Speech Recognition Confidence into Discriminative Named Entity Recognition of Speech Data. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 617–624, Sydney, Australia, July.

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. 2009. Named Entity Recognition from Speech using Discriminative Models and Speech Recognition Confidence. *Journal of Information Processing*, 17:72–81.

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Tsutomu Hirao, and Masaaki Nagata. 2010. Divide and Translate: Improving Long Distance Reordering in Statistical Machine Translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 418–427, Uppsala, Sweden, July.

Unrefereed

須藤 克仁, 東中 竜一郎, 中野 幹生, 相川 清明. 2003. "反省型"信頼性尺度に基づく書き起こしなしデータを用いた言語モデル学習. 情報処理学会研究報告 音声言語情報処理研究会 SLP-45, pages 83–88, February.

須藤 克仁, 中野 幹生. 2004. 音声対話システムのための統計的言語理解モデルの構成とその学習. 言語処理学会第10回年次大会発表論文集, pages 71–74, March.

須藤 克仁, 塚田 元. 2006. 音声対話システムのための音声認識と密に統合した音声言語理解. 日本音響学会春季研究発表会講演論文集, March.

須藤 克仁, 塚田 元, 磯崎 秀樹. 2007. 音声認識の確信度と識別モデルを利用した音声からの固有表現抽出. 第1回音声ドキュメント処理ワークショップ講演論文集. pages 153-158, March.

Katsuhito Sudoh, Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2008. NTT Statistical Machine Translation for IWSLT 2008. In *Proc. of the 5th International Workshop on Spoken Language Translation (IWSLT 2008)*.

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2010. NTT Statistical Machine Translation for IWSLT 2010. In *Proc. of the 7th International Workshop on Spoken Language Translation (IWSLT 2010)*.

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Masaaki Nagata, Xianchao Wu, Takuya Matsuaki, and Jun'ichi Tsujii. 2011. NTT-UT Statistical Machine Translation in NTCIR-9 PatentMT. In *Proceedings of the 9th NTCIR Workshop Meeting*.

須藤 克仁, 永田 昌明, 森 信介. 2012. 日英特許翻訳における日本語単語分割の分野適応の検討. 言語処理学会第18回年次大会発表論文集, pages 1138–1141, March (in Japanese).

須藤 克仁, 進藤 裕之, 塚田 元, 永田 昌明. 2013. 統計翻訳における統語的ラベル細分化の検討. 言語処理学会第19回年次大会発表論文集, pages 390–393, March (in Japanese).

Katsuhito Sudoh, Jun Suzuki, Hajime Tsukada, Masaaki Nagata, Sho Hoshino, and Yusuke Miyao. 2013. NTT-NII Statistical Machine Translation for NTCIR-10 PatentMT. In *Proceedings of the 10th NTCIR Conference*, pages 294–300.

須藤 克仁, 鈴木 潤, 秋葉 泰弘, 塚田 元, 永田 昌明. 2014. 英中韓から日本語への特許文向け統計翻訳システム. 言語処理学会第20回年次大会発表論文集, pages 606–609, March (in Japanese).

Co-authored Works

Refereed

Ryuichiro Higashinaka, Katsuhito Sudoh, and Mikio Nakano. 2006. Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems. *Speech Communication*, 48:417–436.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2012. HPSG-Based Preprocessing for English-to-Japanese Translation. *ACM Transactions on Asian Language Information Processing*, 11(3).

平尾 努, 磯崎 秀樹, 須藤 克仁, Duh Kevin, 塚田 元, 永田 昌明. 2014. 語順の相関に基づく機械翻訳の自動評価法. 自然言語処理, 21(3):421–444, June (in Japanese).

Dan Han, Yusuke Miyao, Pascual Martínez-Gómez, Katsuhito Sudoh, and Masaaki Nagata. 2014. Unlabeled Dependency Parsing Based Pre-reordering for Chinese-to-Japanese SMT. *Journal of Natural Language Processing*, 21(3):485–514, June.

林 克彦, 須藤 克仁, 塚田 元, 鈴木 潤, 永田 昌明. 単語並べ替えと冠詞生成の同時逐次処理：日英機械翻訳への適用. 2014. 単語並べ替えと冠詞生成の同時逐次処理：日英機械翻訳への適用. 自然言語処理, 21(5):1037–1058, September (in Japanese).

Ryuichiro Higashinaka, Katsuhito Sudoh, Mikio Nakano. 2005. Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems, In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010. Head Finalization: A Simple Reordering Rule for SOV Languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251, Uppsala, Sweden, July.

Kevin Duh, Katsuhito Sudoh, Hajime Tsukada, Hideki Isozaki, and Masaaki Nagata. 2010. N-best reranking by multitask learning. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 375–383, Uppsala, Sweden, July.

Katsuhiko Hayashi, Hajime Tsukada, Katsuhito Sudoh, Kevin Duh, and Seiichi Yamamoto. 2010. Hierarchical Phrase-based Machine Translation with Word-based Reordering Model. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 439–446, Beijing, China, August.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic Evaluation of Translation Quality for Distant Language Pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA, October.

Kevin Duh, Katsuhito Sudoh, Tomoharu Iwata, and Hajime Tsukada. 2011. Bayesian Adaptation of Alignment Matrices for Statistical Machine Translation. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Extracting Pre-ordering Rules from Chunk-based Dependency Trees for Japanese-to-English Translation. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Extracting Pre-ordering Rules from Predicate-Argument Structures. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 29–37, Chiang Mai, Thailand, November.

Kevin Duh, Katsuhito Sudoh, Xianchao Wu, Hajime Tsukada, and Masaaki Nagata. 2011. Generalized Minimum Bayes Risk System Combination. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1356–1360, Chiang Mai, Thailand, November.

Kevin Duh, Katsuhito Sudoh, Xianchao Wu, Hajime Tsukada, and Masaaki Nagata. 2012. Learning to Translate with Multiple Objectives. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Jeju Island, Korea, July.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2012. A Comparative Study of Target Dependency Structures for Statistical Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 100–104, Jeju Island, Korea, July.

Dan Han, Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2012. Head Finalization Reordering for Chinese-to-Japanese Machine Translation. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 57–66, Jeju, Republic of Korea, July.

Hirotoshi Taira, Katsuhito Sudoh, and Masaaki Nagata. 2012. Zero Pronoun Resolution can Improve the Quality of J-E Translation. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 111–118, Jeju, Republic of Korea, July.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation Data Selection using Neural Language Models: Experiments in Machine Translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683, Sofia, Bulgaria, August.

Dan Han, Pascual Martínez-Gómez, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2013. Using unlabeled dependency parsing for pre-reordering for Chinese-to-Japanese

statistical machine translation. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 25–33, Sofia, Bulgaria, August.

Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2013. Two-Stage Pre-ordering for Japanese-to-English Statistical Machine Translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1062–1066, Nagoya, Japan, October.

Katsuhiko Hayashi, Katsuhito Sudoh, Hajime Tsukada, Jun Suzuki, and Masaaki Nagata. 2013. Shift-Reduce Word Reordering for Machine Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1382–1386, Seattle, Washington, USA, October.

Dan Han, Pascual Martínez-Gómez, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2013. Analyzing the Influence of Parsing Errors on Pre-reordering Performance for SMT. In *Proceedings of the 27th Pacific Asia Conference on Language, Information and Computation*, November.

Unrefereed

中野 幹生, 東中 竜一郎, Matthias Denecke, 須藤 克仁, 宮崎 昇, 堂坂 浩二. 2004. 対話ログによる訓練が可能なインタラクティブ音声理解システムの枠組. 言語処理学会第10回年次大会発表論文集, pages 67–70, March.

Hideki Isozaki, Katsuhito Sudoh, and Hajime Tsukada. 2005. NTT's Japanese-English Cross-Language Question Answering System. In *Proc. of the 5th NTCIR Workshop Meeting*.

西村 竜一, 秋田 祐哉, 須藤 克仁, 大庭 隆伸. 2006. ICSLPにおける研究動向 - 言語モデル・対話システムを中心に. 第8回音声言語シンポジウム - 情報処理学会研究報告 音声言語処理研究会 SLP-64, pages 239-244, December.

Taro Watanabe, Jun Suzuki, Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. 2007. Larger Feature Set Approach for Machine Translation in IWSLT 2007. In *Proc. of the 4th International Workshop on Spoken Language Translation (IWSLT 2007)*.

堀 貴明, 須藤 克仁, 大庭 隆伸, 渡部 晋治, 渡辺 太郎 , 塚田 元, 中村 篤. 2008. 「世界メディアブラウザ」- 音声認識と統計翻訳に基づく多言語動画コンテンツ検索／閲覧システム. 第2回音声ドキュメント処理ワークショップ講演論文集, pages 59-66, March.

Hideki Isozaki, Tsutomu Hirao, Katsuhito Sudoh, Jun Suzuki, Akinori Fujino, Hajime Tsukada, and Masaaki Nagata. 2009. A Patient Support System based on Crosslingual IR and Semi-supervised Learning. In *Proc. of the SIGIR 2009 Workshop on Information Access in a Multilingual World*.

平尾 努, 磯崎 秀樹, Duh Kevin, 須藤 克仁, 塚田 元, 永田 昌明. 2011. RIBES:語順の相関に基づく機械翻訳の自動評価法. 言語処理学会第17回年次大会発表論文集, pages 1115–1118, March (in Japanese).

CO-AUTHORED WORKS

Shuhei Kondo, Mamoru Komachi, Yuji Matsumoto, Katsuhito Sudoh, Kevin Duh, and Hajime Tsukada. 2011. Learning of Linear Ordering Problems and its Application to J-E Patent Translation in NTCIR-9 PatentMT. In *Proc. NTCIR-9*.

平博順, 須藤 克仁, 永田 昌明 Hirotoshi Taira, Katsuhito Sudoh, and Masaaki Nagata. 2012. An Analysis on Effects of Japanese Zero Pronoun Completion in Statistical Machine Translation. 言語処理学会第18回年次大会発表論文集, pages 135–138, March (in Japanese).

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2012. An Improvement to the Predicate-Argument Structure Based Pre-ordering Approach for Statistical Machine Translation. In *Proceedings of the 18th Annual Meeting of the Association for Natural Language Processing*, pages 151–154, March.

Dan Han, Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2012. Syntactic Based Reordering Rules for Chinese-to-Japanese Machine Translation. In *Proceedings of the 18th Annual Meeting of the Association for Natural Language Processing*, pages 1142–1145, March.

星野 翔, 宮尾 祐介, 須藤 克仁, 永田 昌明. 2013. 日英統計的機械翻訳のための述語項構造に基づく事前並べ替え. 言語処理学会第19回年次大会発表論文集, pages 394–397, March (in Japanese).

星野 翔, 宮尾 祐介, 須藤 克仁, 永田 昌明. 2014. 構文解析誤りに頑健な日英統計的機械翻訳の事前並べ替え手法. 言語処理学会第20回年次大会発表論文集, pages 602–605, March (in Japanese).

Dan Han, Pascual Martínez-Gómez, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2014. Analyzing the Influence of Parsing Errors on Pre-reordering Performance for SMT. In *Proceedings of the 20th Annual Meeting of the Association for Natural Language Processing*, pages 824–827, March.

Chapter 4 of this thesis is the authors' version of the work published in Journal of Natural Language Processing, Volume 21, No. 6, pp. 1107-1131.