NAIST-IS-DD1261016

Doctoral Dissertation

Automated Grammatical Error Correction Using Statistical Machine Translation Techniques with Revision Log of Language Learning SNS

Tomoya Mizumoto

June 18, 2015

Department of Information Science
Graduate School of Information Science
Nara Institute of Science and Technology

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

Tomoya Mizumoto

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Satoshi Nakamura	(Co-supervisor)
Associate Professor Masashi Shimbo	(Co-supervisor)
Assistant Professor Kevin Duh	(Co-supervisor)
Assistant Professor Hiroyuki Shindo	(Co-supervisor)

Automated Grammatical Error Correction Using Statistical Machine Translation Techniques with Revision Log of Language Learning SNS*

Tomoya Mizumoto

Abstract

Recently, natural language processing research has begun to pay attention to second language learning. However, it is not easy to acquire large-scale learners' corpora which are important to a research for second language learner by natural language processing. We present an attempt to extract a large-scale second language learners' corpus from the revision log of a language learning social network service. This corpus is easy to obtain in large-scale, covers a wide variety of topics and styles, and can be a great source of knowledge for both language learners and instructors.

I also demonstrate that the extracted learners' corpus of Japanese/English as a second language can be used as training data for learners' error correction using a statistical machine translation approach. For Japanese error correction, we proposed character-based SMT approach to alleviate the problem of erroneous input from language learners. We evaluate different granularities of tokenization to alleviate the problem of word segmentation errors caused by erroneous input from language learners. Experimental results show that the character-based model outperforms the word-based model.

For English, I conduct experiments in error correction targeting all types errors using statistical machine translation technique and I analyze the strength and weakness of grammatical error correction using statistical machine translation. I also propose two grammatical error correction methods. One is the method considering multi-word expression. Another is the method using discriminative reranking with POS/syntactic features. I show the effectiveness of multi-word expression and reranking for grammatical error correction.

*Doctoral Dissertation, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1261016, June 18, 2015.

Keywords:

Japanese Error Correction, English Error Correction, Language Learning SNS, Revision Log, Character-wise alignment, Multi-word Expression, Discriminative Reranking

語学学習 SNS の添削ログを用いた統計的機械翻訳技術による文法誤り訂正[*]

水本　智也

内容梗概

文法誤り訂正や誤り検出といった自然言語処理による言語学習者支援の研究が盛んに行なわれるようになってきた．しかしながら，自然言語処理による学習者支援を行なう上で重要である大規模な学習者コーパスがないといった問題がある．そこで，本研究では，語学学習 SNS の添削ログから大規模な第 2 言語学習者コーパスの作成を行なった．このコーパスは簡単に大規模に手に入れることができ，さまざまなトピックやスタイルをカバーしており，学習者と教師両方にとって重要な知識源である．

また，これまでの第 2 言語学習者の作文誤り訂正では誤りを限定して行なってきたが，実際の学習者の誤りには様々な誤りが含まれている．そこで，本研究では統計的機械翻訳の手法を誤り訂正に応用して，全ての誤りを対象に日本語/英語の文法誤り訂正を行なった．日本語学習者の文には誤りやひらがなが多く含まれているため，単語分割に失敗してしまうという問題もある．本研究では，単語分割の問題を解決するために文字単位を用いた手法を提案した．文字単位の手法を用いることで，精度が向上することを確認した．

英語に対しても，日本語同様に統計的機械翻訳の手法を用いて文法誤り訂正を行ない，どの誤りタイプに有効であるかを調査した．また，複単語表現を考慮した文法誤り訂正，品詞や構文情報を考慮したリランキングが英語文法誤り訂正に有効であることを示した．

キーワード

日本語誤り訂正, 語学学習 SNS, 添削ログ, 文字単位アラインメント, 複単語表現, リランキング

[*] 奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 博士論文, NAIST-IS-DD1261016, 2015 年 6 月 18 日.

Acknowledgments

忙しい中時間を裂いて研究の相談や進路の相談などに乗ってくれた松本裕治教授に感謝いたします．研究会でのコメント，勉強会の進捗報告に対するコメントなど，本当に有益なコメントを頂きました．

新保仁准教授には，研究会や発表練習などで鋭いご指摘を頂きました．感謝いたします．新保先生の学生の発表スライドに対するコメントは非常に勉強になりました．

Kevin Duh 助教，進藤裕之助教は，自分が博士後期課程に進学してからの付き合いになりますが，博士論文を執筆するにあたり，有益なコメントをいただきました．ありがとうございます．

副指導教員の中村哲教授からも多くの意見を頂きました．ゼミナール，博士論文の発表，博士論文に対するコメントなど大変参考になりました．ありがとうございました．

また，秘書の北川祐子さんにも，出張の手続き，物品購入の手続き等本当にお世話になりました．北川さんがいなければ，事務手続き等でもっと苦しむ学生生活になっていたと思います．本当に感謝しています．

他大に移られて研究室立ち上げで忙しい中でも，親身になって相談に乗ってくれた首都大学東京小町守准教授に感謝いたします．

甲南大学永田亮准教授には学会などであった際に研究の相談に乗っていただいたり，また，フランスでのインターンの機会もいただき感謝しています．自分の力不足でなかなか上手くいかなかったですが，良い経験になりました．

情報通信研究機構藤田篤主任研究員，飯田龍主任研究員には，学会で会った際，研究の話はもちろん進路についてもアドバイスをいただきました．非常に厳しく，辛口なアドバイスですが，心に刻んでこれからも頑張っていきたいと思います．松本研 OB ですが，世代的に被っていない中いろいろと話していただき，感謝しています．

NAIST での 5 年間，研究，生活を行なうにあたって同期，先輩，後輩問わずいろいろな人に助けられてきました．ありがとうございました．

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The number of second language learners has incresed. The number of Japanese language learners around the world has increased by more than 30-fold in the past three decades (Figure 1.1). The Japan Foundation reports that more than 3.65 million people in 133 countries and regions are studying Japanese in 2009[1]. However, there are only 50,000 Japanese language teachers overseas (Figure 1.2), and thus it is in high demand for finding good instructors for learners of Japanese as a Second Language (JSL). The learners of English as a Second Language (ESL) also has increased. Figure 1.3 shows the number of test takers for International English Language Testing System (IELTS). IELTS is one of the international tests of English language proficiency for non-native English speakers. The test takers are increased by about 1.5 million people for six years.

Publicly usable services on the Web for assisting second language learning are growing recently. For example, there are language learning social networking services such as Lang-8[2] and English grammar checkers such as Ginger[3] and 1checker[4]. Research on assistance of second language learning also has received much attention, especially on grammatical error correction of essays written by learners of English as a second language (ESL) . In the past, four competitions for grammatical error correction have been held: Helping Our Own (Dale and Kilgarriff, 2011; Dale et al., 2012) and CoNLL Shared Task (Ng et al., 2013; Ng et al., 2014).

Recently, natural language processing research has begun to pay attention to second

[1] `http://www.jpf.go.jp/e/japanese/survey/result/index.html`

[2] `http://lang-8.com`

[3] `http://www.gingersoftware.com`

[4] `http://www.1checker.com/`

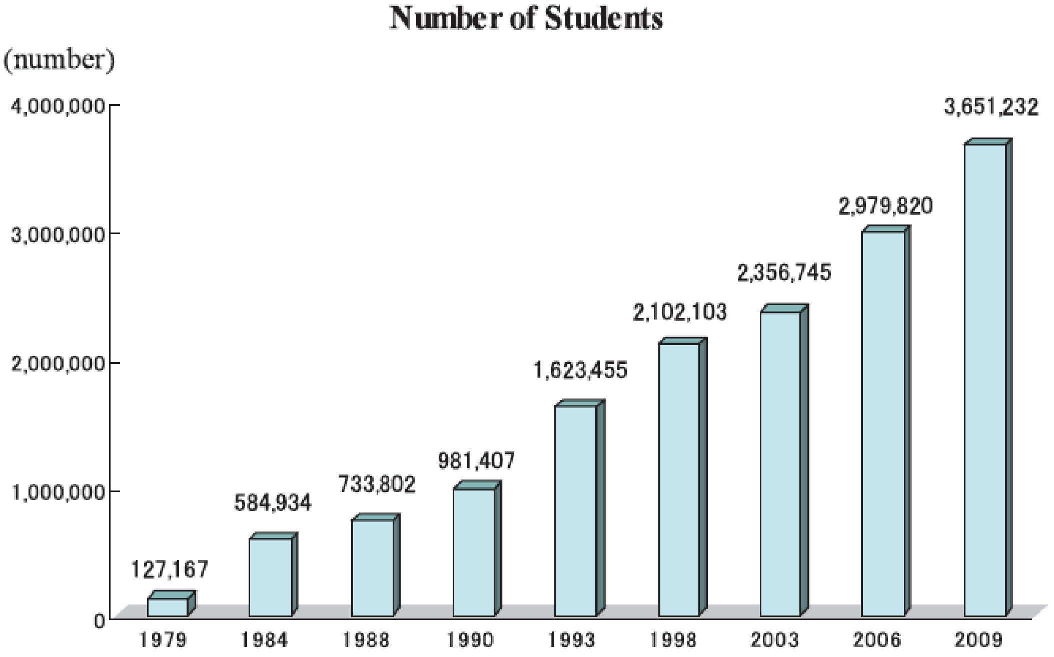


Figure 1.1: Number of students studying Japanese overseas [Source: The Japan Foundation; Present Condition of Overseas Japanese-Language Education Survey Report on Japanese-Language Education Abroad 2009]

language learning (Rozovskaya and Roth, 2011; Park and Levy, 2011; Liu et al., 2011; Oyama and Matsumoto, 2010; Xue and Hwa, 2010). Most previous research on language learners' grammatical error correction is targeted on one or few restricted types of learners' errors. For example, research for JSL learners' errors mainly focuses on Japanese case particles (Oyama and Matsumoto, 2010; Imaeda et al., 2003; Nampo et al., 2007; Suzuki and Toutanova, 2006) and for ESL learners' errors mainly focuses on articles and preposition. However, real JSL learners' writing contains not only errors of Japanese case particles but also various other types of errors including spelling and collocation errors. For instance, a Japanese language learner who speaks Chinese may write:

> 何で日本語はこんなに難しいなの？
> (Why is Japanese so difficult?)

which has a grammatical error of inserting 'な' because isolating language (such as Chinese) speakers find it hard to learn how to use adjective conjugation forms correctly. Park and Levy (2011) proposed an EM-based unsupervised approach to perform whole

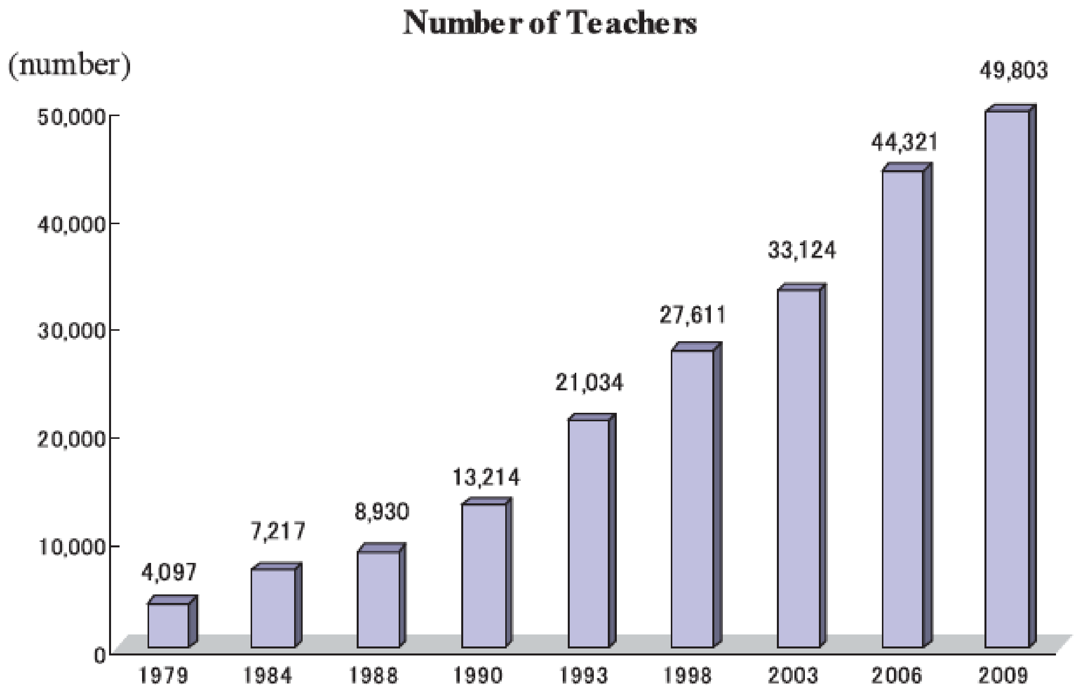


Figure 1.2: Number of Japanese language teachers overseas [Source: The Japan Foundation; Present Condition of Overseas Japanese-Language Education Survey Report on Japanese-Language Education Abroad 2009]

sentence grammar correction, but the types of errors must be pre-determined to learn the parameters for their noisy channel model. It requires expert knowledge of second language (L2) teaching, which is often hard to obtain.

One promising approach for correcting unrestricted errors of second language learners is Brockett et al. 's automated error correction method (Brockett et al., 2006) using statistical machine translation (SMT). The advantage of their method is that it does not require expert knowledge. Instead, it learns a correction model from sentence-aligned corrected learners' corpora. However, it is not easy to acquire large-scale learners' corpora. In fact, Brockett et al. (2006) used regular expressions to automatically create erroneous corpora from native corpora.

To alleviate the knowledge acquisition bottleneck, I propose a method of mining revision logs from a language learning social network service (SNS) to create a large-scale learners' corpus. The SNS covers a wide variety of topics and styles. The main advantage of using revision logs from SNS is three-fold: (1) it benefits from the wis-

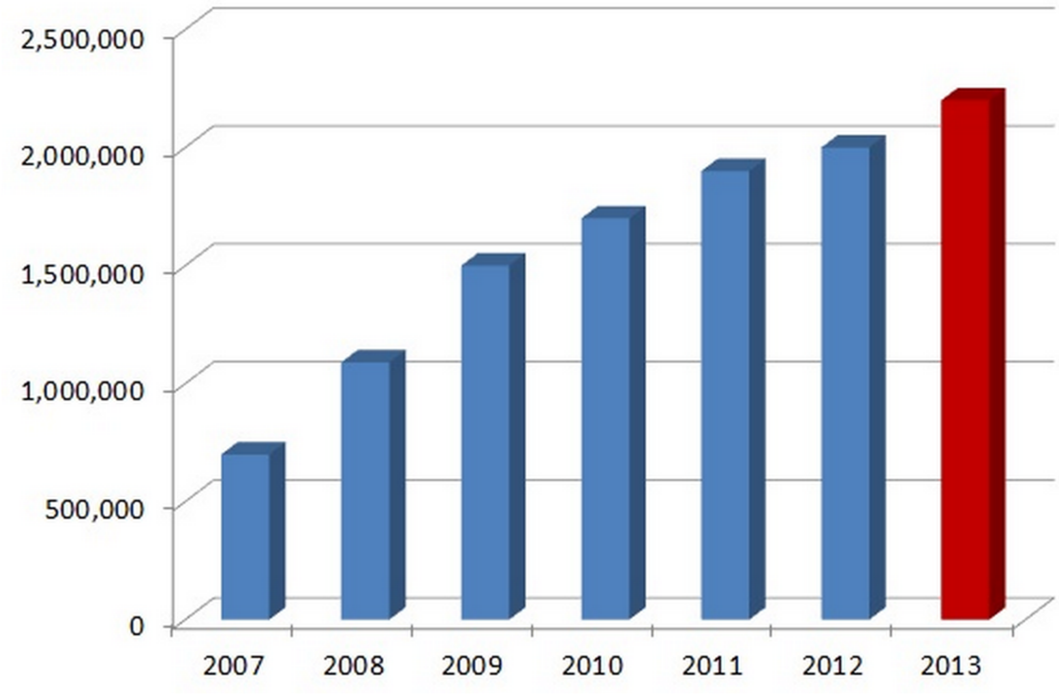


Figure 1.3: Number of test takers for IELTS [Source: Situation of examination; http://www.eiken.or.jp/ielts/merit/situation/]

dom of crowds, (2) logs can be obtained on a large scale, and (3) logs are a great source of knowledge not only for learners but also for language teachers.

In this thesis, I show that the method using phrase-based SMT technique with a large-scale learners' corpus can correct second language learners' errors with reasonable accuracy. I also show that considering multi-word expressions and reranking with syntactic informatrion for SMT outputs is effective for grammatical error correction.

The rest of this thesis is organized as follows. Chapter 2 describes the JSL and ESL corpus created from revision logs of a language learning SNS. Chapter 3 explains an SMT-based approach to grammatical error correction for second language learner. In Chapter 5, I report the experimental results of SMT-based JSL error correction using a large-scale real corpus. In Chapter 6, I show the effect of learner corpus size on the SMT approach. Chapter 7 shows the effects of using multi-word expressions for grammatical error corection. In Chapter 8, I report the experimental results of reranking approach to gramamtical error correction. In Chapter 9, I concludes this work.

Chapter 2

A Large Scale Japanese Language Learners' Corpus from Revision Logs of Language Learning Social Network Service

There are already many language learners' corpora. However, there is not a large-scale learner corpus on freely available. In Section 2.1, I show the Japanese and English language learner corpora.

2.1 Language Learners' Corpora

2.1.1 Japanese Language Learners' Corpora

One of the most well-known Japanese learners' corpus is Teramura Error Data[1]. The corpus was mainly collected in 1986 from Japanese compositions written by foreign students, mostly from Asian countries. The corpus consists of several styles including writing exercises, cloze (gap filling) test, and pattern composition. Unlike this data, JSL learners in Lang-8 encompass the whole world. Also, Lang-8 offers a wide variety of free compositions of the learner's choice, and the size of the data is 3 orders of magnitude (448MB without all the tags) larger than Teramura's data (420KB, 4,601 sentences written by 339 students). Also, although Teramura Error Data is annotated

[1] http://www.ninjal.ac.jp/teramuragoyoureishu/

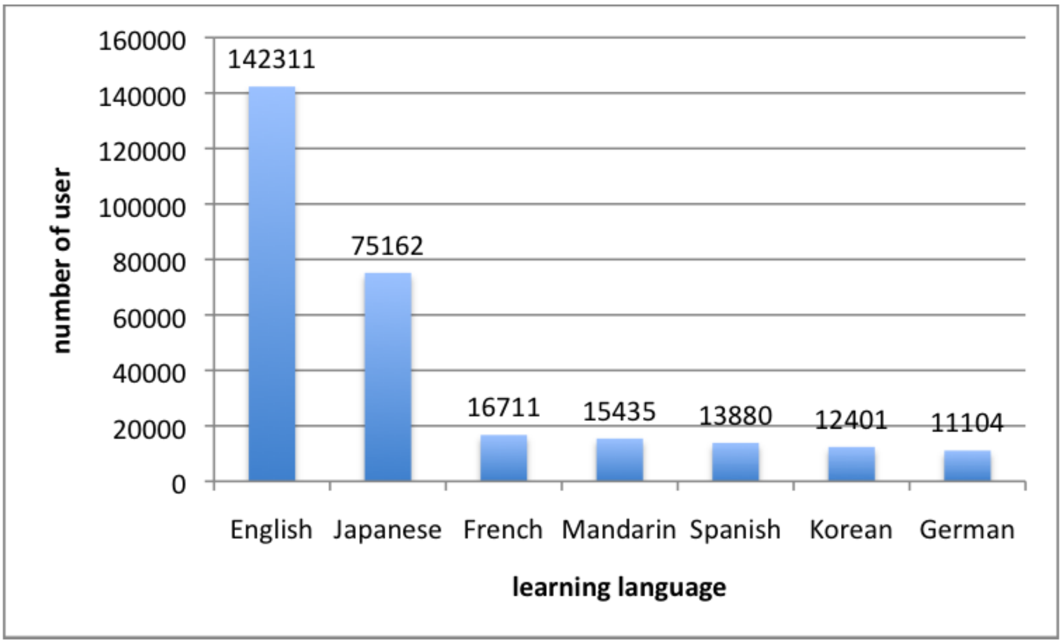


Figure 2.1: Number of users for each learning language in Lang-8

with error types, the correct words or strings are not often provided, which makes it difficult to use it for automatic correction of learners' errors.

Ohso[2] created a database of Japanese compositions by JSL learners. It is annotated with error types with correct forms to allow error analysis. However, similar to Teramura Error Data, the corpus does not cover many topics because it was collected at only four institutions. In addition, it is limited in size (756 files, average file size is 2KB).

The corpus most related to mine is the JSL learners parallel database of Japanese writings and their own translation into their first language[3] created by National Institute for Japanese Language and Linguistics. It collects 1,500 JSL learners' writings and their self-translations. There are around 250 writings corrected by several Japanese language teachers. The advantage of this corpus is that some of the texts are annotated by professional language teachers and can be used as a source of error correction. However, again, the size of this corpus is limited since it is hard to obtain annotations from language teachers. My approach differs from them in that I employ the wisdom of crowds of native speakers, not necessarily language teachers, to compile a large-scale

[2] https://kaken.nii.ac.jp/ja/p/08558020/1998/6/en
[3] http://jpforlife.jp/taiyakudb.html

Table 2.1: Comparison of English language learner corpus.

Name of Leaner corpus	Data size
CLC	over 200,000 texts
CLC-FCE	1,244 texts
NICT JLE Corpus	1.2 million words
Konan-JIEM Corpus	233 texts, 3,199 sentences
NUCLE	1,397 essays, 57,151 sentences
CoNLL-ST13 Test Set	50 essays, 1,381 sentences
CoNLL-ST14 Test Set	50 essays, 1,312 sentences
Lang-8 Corpus	1,069,549 sentences

learners' corpus.

2.1.2 English Language Learner Copora

There are many English language learner copora compared to Japanese language corpora. I only compare the well-known English language learner corpora which errors are annotated.

Table 2.1 shows comparison of English language learner corpus. The most well-known and biggest English Corpus is Cambridge Learner Corpus (CLC) (Nicholls, 2003), but this is not freely available. Cambridge ESOL First Certificate in English (CLC-FCE) is a subset of CLC, which contains only 1,244 texts. NICT JLE Corpus (Izumi et al., 2004), Konan-JIEM Corpus (Nagata et al., 2011) and NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) are available research purpose, however these corpora are too small. The test sets of CoNLL Shared Task 2013 (Ng et al., 2013) and 2014 (Ng et al., 2014) are also very small.

2.2 Language Learning Social Network Service

Recent growth of the web has opened the possibility of using the Internet to break the barriers of space and time. Specifically, social network service (SNS) has begun to receive a lot of attention recently. There are a number of SNS sites that help language learners across the world, including iKnow!, Livemocha and Lang-8, to name a few. I will look briefly at each SNS below.

Table 2.2: Number of sentences for each language in Lang-8

Language	English	Japanese	Mandarin	Korean	Spanish	French	German
Number of sentences	1,069,549	925,588	136,203	93,955	51,829	58,918	37,886

First, iKnow![4] is an SNS-based language learning service that helps learners practice language learning. iKnow provides a tailored curriculum for each user to memorize words and phrases through simple exercises.

Second, Livemocha[5] is also a language learning SNS that offers courses of grammar instructions, reading comprehension exercises and practice for both writing and speaking. It provides educational materials in 38 languages. Users can submit a writing exercise on a subject and receive feedbacks from other users of the native language.

Third, Lang-8 is a “Multi-lingual language learning and language exchange Social Networking Service”[6] , which has 214,170 (317,307) registered members as of November, 2010 (October, 2011). Soon after the learners write a passage, mostly a part of a diary, in a language they are learning, native speakers of the language correct it for them. The learners in turn are encouraged to correct other members’ composition errors according to their first language (L1). Hence, the SNS is called “language exchange”. It supports 77 languages, facilitating multilingual communication. In this chapter, I use the data of Lang-8 crawled at November, 2010 for Japanese learner corpus and September, 2011 for English learner corpus[7].

2.3 Features of Lang-8 Data

I created a large-scale language learners’ corpus from error revision log of Lang-8. Figure 2.1 shows that approximately 75,000 users are learning Japanese[8]. Table 2.2 shows the top seven languages in the corpus. There are 925,588 sentences of JSL learners[9]. Out of 925,588 sentences, 763,971 (93.4%) sentences are corrected by

[4] `http://iknow.jp`

[5] `http://www.livemocha.com/`

[6] `http://lang-8.com/`

[7] Recent data (after October 2011) is not free because Lang-8 changed their policy on the 6th October 2011 so that you need permission from the site maintainer to use the data

[8] I counted learning language in user profile. Some learners register two or more learning languages.

[9] I counted learning language written for each journal because learners may write in different languages.

Table 2.3: An illustrative example of multiple correction

Sentence written by a JSL learner	三人はそれぞれ自分の方式で感情を表れます。
Sentence corrected by an annotator1	三人はそれぞれ自分なりの表現で感情を表します。 (Each of three expresses their feelings in their own expressions.)
Sentence corrected by an annotator2	三人はそれぞれ自分なりに感情を表します。 (Each of three expresses their feelings in their own way.)

human annotators. A sentence written by JSL learners might have two or more revision sentences in Lang-8 by different voluntary reviewers[10]. Therefore, the total number of corrected sentences amounts to 1,288,934. In other words, one sentence gets corrected approximately 1.69 times on average.

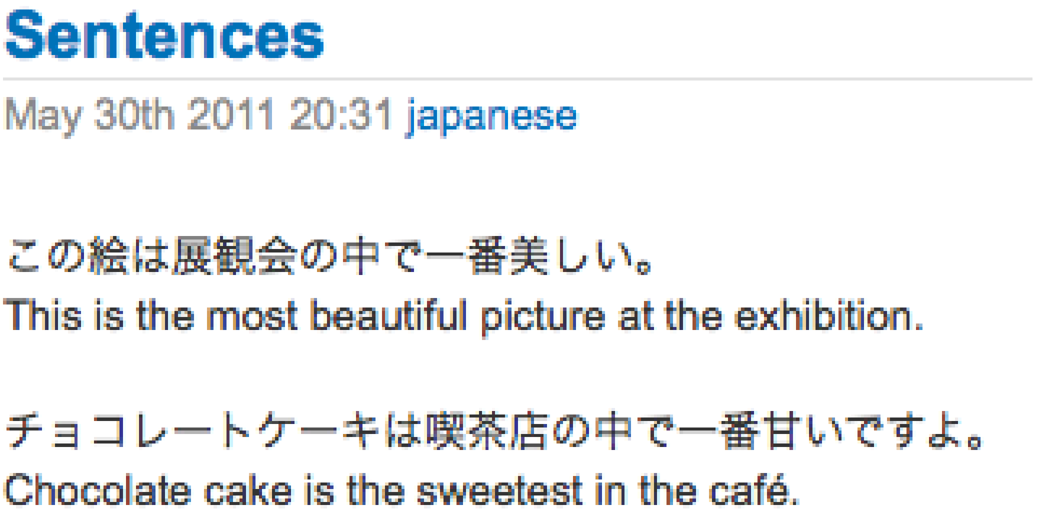
There are several distinguishing features of the data obtained from Lang-8. First, since Lang-8 is a language learning SNS, I can obtain pairs of learner's sentence and corrected sentence. Using this data, it is possible to collect the learners' errors. I will describe how to build a learners' corpus from revision logs later in this section.

Second, Lang-8 data may have more than one correction for the single sentence. I could exploit this feature to acquire paraphrases in a similar way to (Barzilay and McKeown, 2001). Table 2.3 shows an example of multiple correction. Two annotators correct the same learner's sentence. In this example, one can infer that "なりの表現で (in one's own expressions)" and "なりに (in one's own way)" are paraphrases of each other.

Third, I could obtain multi-lingual parallel sentences. Figure 2.2 shows examples of parallel sentences in Lang-8. In this example, the JSL learner writes two Japanese sentences and their translation for each sentence to tell what he or she wants to say. Although the sentences written in the learning language may contain errors and mistakes, I can align the English translation to the corrected Japanese sentence. The parallel corpus created from the revision log of SNS would be a valuable source of colloquial expressions ideal for translating consumer generated media such as blogs and SNS.

Fourth, annotators of Lang-8 sometimes add inline comments to the corrected sentences. It is often written in parentheses to indicate that the string is a comment, but not always. Depending on the first language of the language learner, annotators put comments in either the learning language or the learner's L1. This can be a great source of

[10] The correction of a new review might be affected by the previous corrections by others.



Sentences

May 30th 2011 20:31 japanese

この絵は展観会の中で一番美しい。
This is the most beautiful picture at the exhibition.

チョコレートケーキは喫茶店の中で一番甘いですよ。
Chocolate cake is the sweetest in the café.

Figure 2.2: Parallel sentence in Lang-8

extracting useful information for language learning, since the comment itself explains pitfalls that the language learners often come across.

2.4 Extracting Corrected Sentences from HTML

All the error revisions are made through a web-based editing interface that allows annotators to delete, insert or change any character sequence of the learner's text by any sequence. Table 2.4 illustrates an example of the HTML generated from Lang-8's revision editor. The tag `<span class="sline">` shows that the characters within the tags should be removed. The color tags `<span class="red">` and `<span class="f_blue">` are used somewhat arbitrarily by annotators. In general, they indicate correct strings. In the example, the annotator used delete line and red color to point out and correct the first error, and blue color to indicate inserted characters.

From this observation, I apply simple heuristics to extract corrected sentences from Lang-8. First, I remove all the `<span class="sline">` tags and characters within them. Then, I discard other tags, retaining the characters surrounded by the tags. After this rule, I obtain the corrected sentence shown in the bottom row in Table 2.4.

Table 2.4: Extracting corrected sentence from HTML

Sentence written by a JSL learner	去年は参加してなかった、見るだけ。 (I was not participating last year, just watching.)
Corrected sentence with tags	去年は参加<span class="sline">してなかった</span><span class="red">せずに</span>、見るだけ<span class="f_blue">だった</span>。
Seen on the browser	去年は参加~~してなかった~~せずに、見るだけだった。
Corrected sentence	去年は参加せずに、見るだけだった。 (I did not participate but watched last year.)

2.5 Data Statistic and Filtering by Edit Distance

In actual correction, it is expected that annotators do not completely rewrite the original sentence and most character strings remain the same as the original sentence. Thus, I investigated the quantitative distribution of Lang-8 data by breaking down the sentences according to the edit distance between the original and corrected sentences (number of deletion / number of insertion of characters in revision log).

Figures 2.3 and 2.4 summarize the numbers of deleted and inserted characters in a sentence. These figures show that two distributions are comparable. On the other hand, they differ in the absolute number of deletion and insertion. For example, the number of cases with no deletion is considerably higher than the number with no insertion. Also, the frequency of sentences with more than nine insertions is higher than that for deletions. This reflects the fact that there are many sentences with comments (insertions) and that people tend not to remove too many characters to keep the information of the original sentence written by the learner.

From observations of the created corpus, corrections can be divided into two types: (1) a correction by insertion, deletion, or substitution of strings, (2) a correction with a comment. Table 2.5 shows examples of correction from Lang-8. The first example is a sentence written by a JSL learners containing an error, and is corrected by inserting a character. In the second example the learner's sentence is correct; in addition the annotator writes a comment[11]. Besides, there exist "corrected" sentences to which only the word "GOOD" is appended at the end. In this case, original sentence is not modified at all by the annotator. The inserted comment merely informs the learner that there is no mistake in the learner's writing.

[11] Some annotators erase a learner's original sentence and rewrite it to "OK".

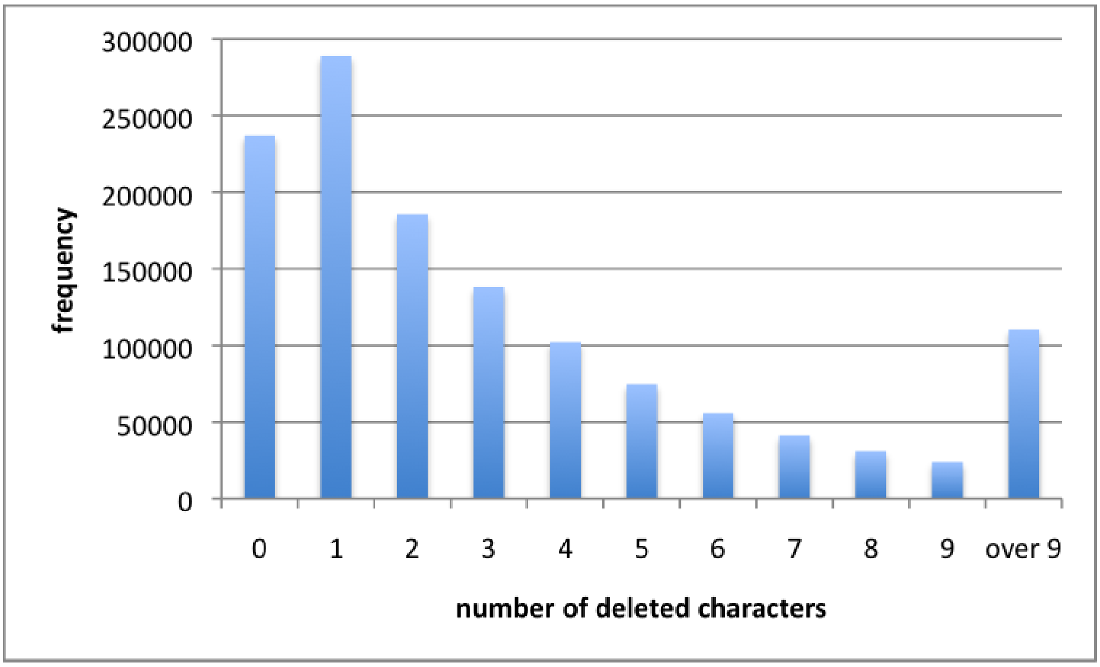


Figure 2.3: Summary of number of deletion

To handle these comments, I conduct the following three pre-processing steps: (1) if the corrected sentence contains only “GOOD” or “OK”, I do not include it in the corpus, (2) if edit distance between the learner’s sentence and corrected sentence is larger than 5, I simply drop the sentence for the corpus, and (3) if the corrected sentence ends with “GOOD” or “OK”, I remove it and retain the sentence pair. As a result, I obtained a corpus of 849,894 corrected and aligned sentence pairs by JSL learners.

Another notable issue is that annotators may not correct all the errors in a sentence. Table 2.6 shows an example of JSL learner’s sentence for confusing case markers of “が” (NOM) and “は” (TOP). In this example, “は” and “が” should be corrected to “が” and “は”, respectively. However, the annotator left the second case markers “は” unchanged. Because the number of these cases seems low, I regard it as safe to ignore this issue for creating the corpus.

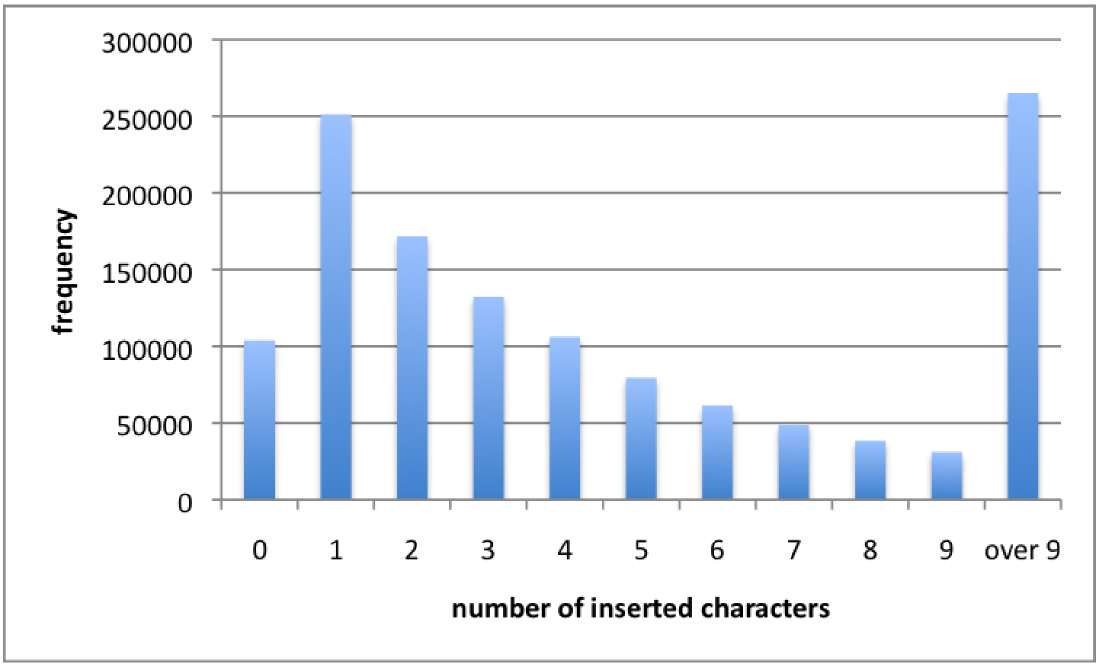


Figure 2.4: Summary of number of insertion

Table 2.5: Examples of correction in Lang-8

Sentence written by a JSL learner	ビデオゲームをやました (Video games Yamashita.)
Sentence corrected by an annotator	ビデオゲームをやりました (I played video games.)
Sentence corrected by a JSL learner	銭湯に行った。 (I went to a public bath.)
Sentence corrected by an annotator (with comment)	銭湯に行った。 いつ行ったかがある方がいい (I went to a public bath. It is better to say when you went.)

Table 2.6: Problem of correction in Lang-8

Sentence written by a JSL learner	この４つが僕は少年のころに発売されて (As for me, these four were sold when I was a kid.)
Sentence corrected by an annotator	この４つは僕は少年のころに発売されて (As for these four, I was sold when I was a kid.)
Correct sentence	この４つは僕が少年のころに発売されて (As for these four, they were sold when I was a kid.)

Chapter 3

Grammatical Error Correction using Phrased-based Statistical Machine Translation Methods

In this thesis, I attempt to solve the problem of second language learners' error correction using the Statistical Machine Translation (SMT) technique. Brown et al. (1993) first proposed word-based SMT approach. While the common SMT problem is a task translating from source language to target language, i.e., English to Japanese (Figure 3.1) grammatical error correction can be considered as a task of translating from incorrect sentences to correct sentences (Figure 3.2).

Related work on grammatical error correction using phrase-based SMT includes research on English and Japanese (Brockett et al., 2006; Suzuki and Toutanova, 2006). Brockett et al. (2006) proposed to correct mass noun errors using SMT and used 45,000 sentences as training sets randomly extracted from automatically created 346,000 sentences. My work differs from them in that I (1) do not restrict myself to a specific error type such as mass noun; and (2) exploit a large-scale real world data set.

The use of SMT for spelling and grammar correction has the following three advantages.

1. It does not require expert knowledge.

2. It is straightforward to apply SMT tools to this task.

3. Error correction using SMT can benefit from the improvement of SMT method.

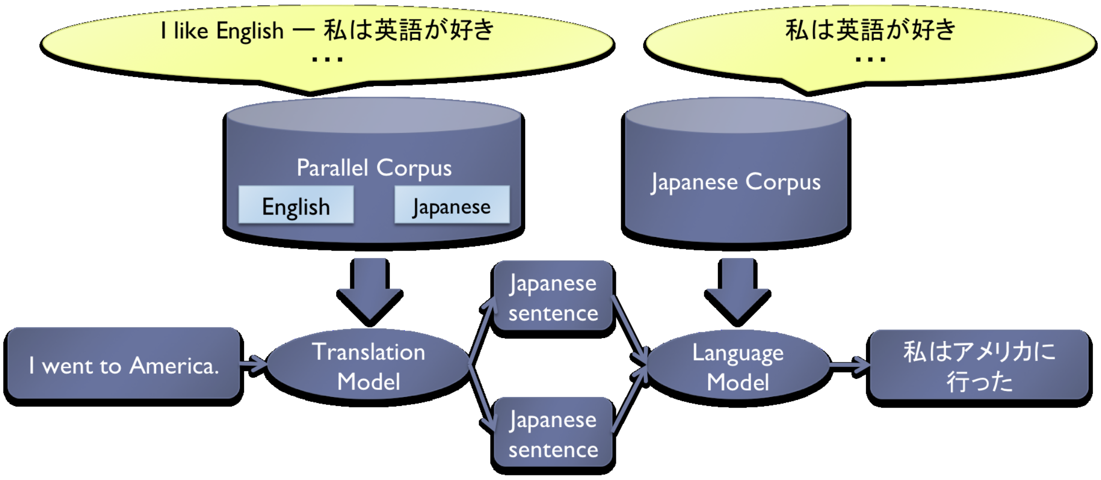


Figure 3.1: Example of common translation problem.

3.1 Statistical Machine Translation Formulation Using a Log-Linear Model

The well-known statistical machine translation formulation using a log-linear model (Och and Ney, 2002) is defined by:

$$\hat{e} = \arg\max_{e} P(e|f) = \arg\max_{e} \sum_{m=1}^{M} \lambda_m h_m(e,f) \tag{3.1}$$

where e represents target sentences (corrected sentences) and f represents source sentences (sentences written by learners). $h_m(e,f)$ is a feature function and λ_m is a model parameter for each feature function. This formulation finds a target sentence e that maximizes a weighted linear combination of feature functions for source sentence f. A translation model and a language model can be used as feature functions.

The translation model is commonly represented as conditional probability $P(f|e)$ factored into the translation probability between phrases. The language model is represented as probability $P(e)$. The translation model is learned from sentence-aligned parallel corpus while the language model is learned from target raw corpus.

3.2 Phrase Extraction

Phrases are extracted from a parallel corpus which word alignment is annotated with popular heuristics in SMT task (Och and Ney, 2003). As preprocessing for phrase ex-

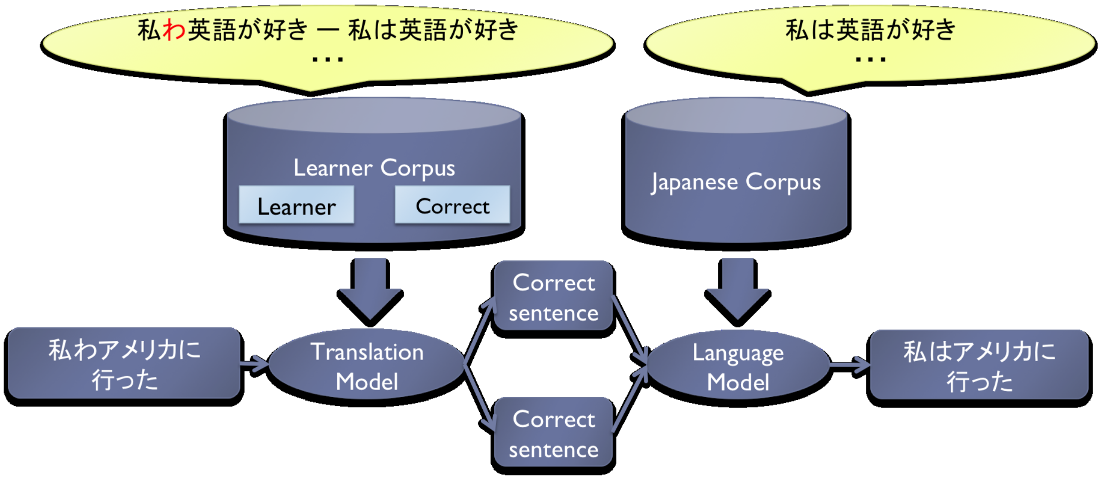
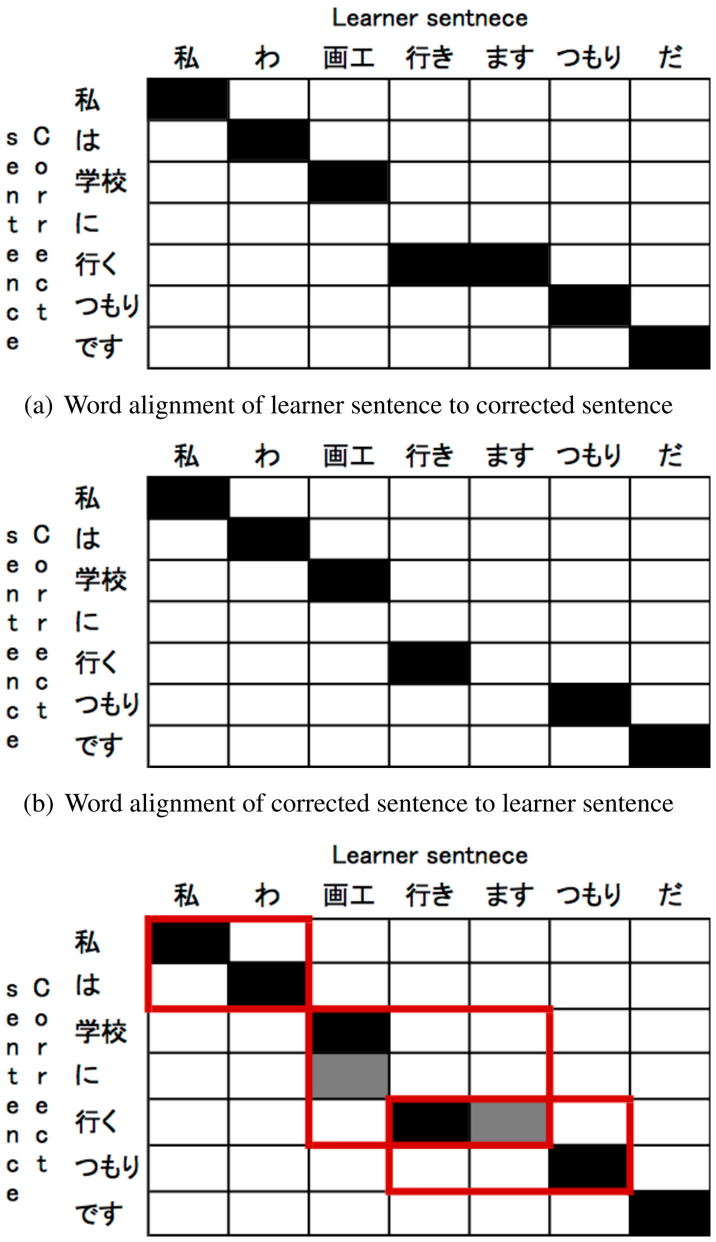


Figure 3.2: Example of grammatical error correction using SMT.

traction, both direction of word alignment of learner sentence to corrected sentence and word alignment of corrected sentence to learner sentence are conducted. Figure 3.3(a) shows the result of word alignment of learner sentence to corrected sentence. Figure 3.3(b) shows the result of word alignment of corrected sentence to learner sentence. On the heuristics for phrase extraction; (1) the alignments in the intersection set of the both direction of word alignments are first added, (2) and then neighboring alignment points in the union sets are added, (3) and then the diagonally neighboring alignment points are also added, (4) and finally, the non-neighboring alignment points between words, of which at least one is currently unaligned, are added.

The black points in Figure 3.3(c) are intersection of the both direction of word alignments. The gray points in Figure 3.3(c) are added by heuristics for phrase extraction. The phrases pair that all words of the phrase pair have to align to each other are extracted. The areas enclosed by red line in Figure 3.3(c) are part of extracted phrase. For example, the following phrases can be extracted; “私わ” aligned to “私は”, “画工行きます” aligned to “学校に行く” and “行きますつもり” aligned to “行くつもり”.



(a) Word alignment of learner sentence to corrected sentence

(b) Word alignment of corrected sentence to learner sentence

(c) Phrase Extraction

Figure 3.3: Phrase extraction from word alignment

Chapter 4

Evaluation Metrics

For the evaluation metrics, I use automatic evaluation criteria. To be precise, I used recall (R), precision (P) , F1-Score and F0.5-Score. Recall, precision, $F1_1$-Score and $F_{0.5}$-Score are defined as follows:

$$Recall = \frac{tp}{tp+fn},$$

$$Precision = \frac{tp}{tp+fp},$$

$$F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision},$$

$$F_{0.5} = \frac{(1+0.5^2) \times Recall \times Precision}{Recall + 0.5^2 \times Precision}$$

where tp (true positive), fp (false positive), and fn (false negative) denote the case that the system correctly identifies learner's errors, the case that the system incorrectly identifies learner's errors, and the case that the system fails to identify learner's errors, respectively. F_1-Score is the harmonic mean of recall and precision. $F_{0.5}$-Score weights precision twice as much. To illustrate recall and precision, let us consider the example in Figure 4.1. tn in Figure 4.1 counts the cases that all the learner's input, system output and gold standard are the same. The numbers of tp, fp and fn in Figure 4.1 are 1, 1 and 2, respectively. Thus, $recall = 1/3$ and $precision = 1/2$.

In Chapter 6, I calculate the scores for each error types. Recall and precision for each type of errors are calculated from true positive, false positive and false negative based on error tags in evaluation corpus. The word which does not have any tag in

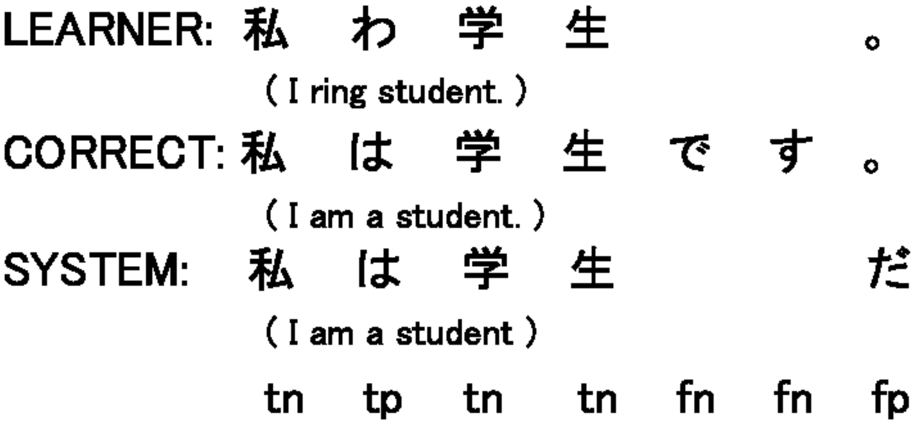

LEARNER: 私 わ 学 生 。
（I ring student.）
CORRECT: 私 は 学 生 で す 。
（I am a student.）
SYSTEM: 私 は 学 生 だ
（I am a student）

Figure 4.1: Example of evaluation

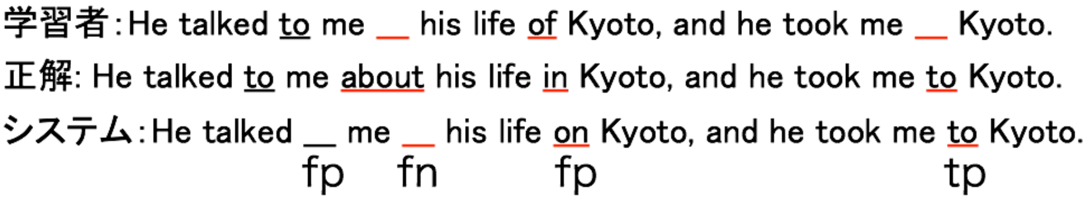

学習者:He talked to me __ his life of Kyoto, and he took me __ Kyoto.
正解: He talked to me about his life in Kyoto, and he took me to Kyoto.
システム:He talked __ me __ his life on Kyoto, and he took me to Kyoto.

Figure 4.2: Example of evaluation for each error types

evaluation corpus does not affect precision for each type of errors[1]. For example, let us consider the following: In this example, the system deletes preposition “to”, which does not have any tag. Thus, precision = 1/2, recall = 1/2 for *preposition* errors and precision = 1/3, recall = 1/2 for Total scores.

[1] The total score is calculated using all the correction output with and without any tag.

Chapter 5

Japanese Error Correction Using Character-wise Word Alignment

5.1 Statistical Error Correction with Different Granularity of Tokenization

When translating a sentence from Japanese to another language with SMT, one usually performs word segmentation as a pre-processing step. However, JSL learners' sentences contain a lot of errors and hiragana (phonetic characters), which are hard to tokenize by traditional morphological analyzer trained on standard Japanese writings. Suppose I want to tokenize the following real sentence written by a JSL learner:

でもじょずじゃりません

where the correct counterpart would be:

でもじょうずじゃありません
(But I am not good at it.)

The corrected sentence has "う" and "あ" inserted[1]. These sentences written by a learner and corrected by a native speaker are tokenized as follows by MeCab[2], which is one of the most popular Japanese Morphological Analyzer:

でも	じ	ょずじゃりません
(but	(fragment)	(garbled word))

でも	じょうず	じゃ	あり	ませ	ん
(but	good	at	be	not)	

[1] It is hard for JSL learners of certain L1 to distinguish Japanese short and long vowels.

[2] http://mecab.sourceforge.net/

These examples illustrate the difficulty of correcting JSL learners' sentence using word-wise SMT.

To alleviate this problem, I propose to build a character-wise segmented corpus with phrase-based SMT. The Character-wise model is not affected by word segmentation errors, and thus it is expected to be more robust for the task of correcting JSL errors. For the two example sentences mentioned above, I split sentences into characters rather than words:

で も じ ょ ず じ ゃ り ま せ ん

で も じ ょ う ず じ ゃ あ り ま せ ん

This enables the phrase-based SMT to learn the alignment between "じょず" and "じょうず" (Figure. 5.1), resulting in a more robust model to correct JSL errors than word-wise model.

Moreover, I propose a combined method in which the source language is tokenized character-wise while the target language is tokenized word-wise (Figure. 5.2). The intuition behind this is that the source language (sentence written by learners) is hard to tokenize into words, whereas the target language (corrected sentences) may be easy to tokenize.

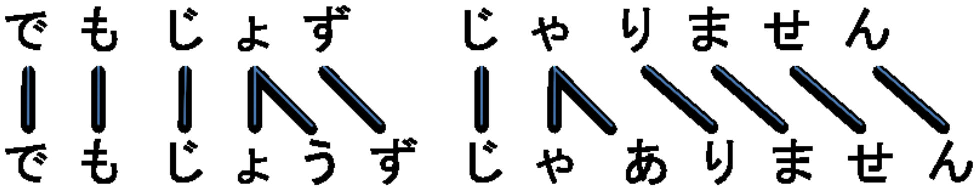

で も じ ょ ず じ ゃ り ま せ ん
で も じ ょ う ず じ ゃ あ り ま せ ん

Figure 5.1: Example of character alignment

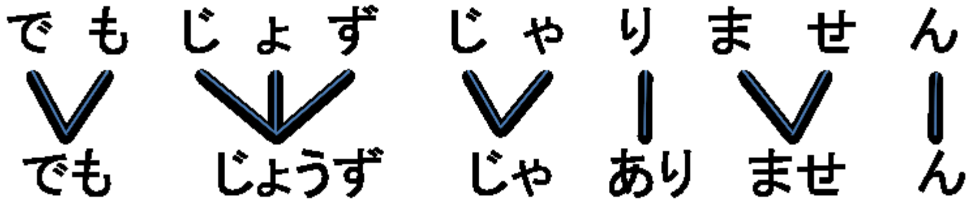

で も じ ょ ず じ ゃ り ま せ ん
でも じょうず じゃ あり ませ ん

Figure 5.2: Example of character - word alignment

5.2 Experiments on JSL Learner's Error Correction with SMT

I carried out experiments to see (1) the effect of granularity of tokenization as described in Section 5.1; (2) the effect of corpus size; (3) the difference of L1 model. I also carried out experiments using NAIST Goyo corpus[3] to see (a) effectiveness of Lang-8 corpus, (b) effectiveness for each error type.

I used Moses 2010-08-13[4] as an SMT tool and GIZA++ 1.0.5[5] as an alignment tool. I used Japanese morphological analyzer MeCab 0.97 with UniDic 1.3.12[6] for word segmentation.

I created a word-wise model as baseline. Hereafter, I refer to this as W-W and also constructed a model with entries from UniDic for better alignment, denoted as W-W+D. I used word 3-gram as language model for W-W and W-W+D. I built two character-wise models: Character 3-gram and 5-gram represented as C-C3 and C-C5, respectively. Also, I created a combined model of word and character. In particular, I apply character-wise segmentation on the source side and word-wise segmentation on the target side. Hereafter, I refer to this as C-W. As before, I prepared a model with entries from UniDic, denoted as C-W+D. I conducted minimum error rate training (MERT) (Och, 2003) in all experiments. I performed minimum error rate training to maximize BLEU (Papineni et al., 2002) (5-gram).

5.2.1 Experimental Data

All the data was created from 849,894 Japanese sentences extracted from revision logs of Lang-8 crawled in December 2010. I retained pairs of sentences whose number of characters of corrected sentence is less than or equal to 50. This results in 796,956 sentences out of 849,894 sentences. To see the difference of errors stemming from learners' L1, I carried out an experiment with two L1s: English and Mandarin. ALL extracts training data from the entire corpus for the translation model. There are 298,359 Japanese sentences whose writers' L1 is English and 166,688 Mandarin. For

[3] NAIST Goyo corpus is annotated error types to JSL learners parallel database of Japanese.

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

[5] http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/GIZA++.html

[6] http://www.tokuteicorpus.jp/dist/

each L1 JSL corpus, I split the corpus into two parts: 500 sentences for testing and development, and the rest for training.

I shuffled the training data to prepare the corpus for learning language model and translation model. I manually re-annotated 500 sentences to make gold-standard data and used 200 sentences for testing, and 300 sentences for development.

NAIST Goyo corpus consists of 6,433 sentences. I split the corpus into three parts: 5,933 sentences for training, 300 sentences for testing, and 200 sentences for development.

5.2.2 Experimental Results for Test Data from Lang-8

Comparison of granularity of tokenization Table 5.1 shows the performance with different methods in the cases where test data were written by English and Mandarin speakers (Training Corpus: L1 = ALL; translation model: 300K sentences; language model: 790K sentences). The character-wise models and the combined models outperform the word-wise models in the case where test data was written by English speakers, while word-wise model outperform the character-wise model and the combined models in the case where test data was written by Mandarin speakers. C-W achieved the best recall and F-measure and C-C5 the best precision in the case where test data was written by English speakers. W-W achieved the best recall and F-measure and W-W+D the best precision in the case where test data was written by Mandarin speakers.

Effects of corpus size I varied the size of the corpus used to train translation models while fixing the size of the training corpus of language models to 790M sentences in order to see the effect of the size of training corpus on the performance. Figures 5.4, 5.3 and 5.5 show the performance with different size of training corpus of translation model, using W-W, C-C5 and C-W (Training Corpus: L1 = ALL). In all configurations, the best F-measure was achieved when the size of training corpus of translation model was larger size.

Comparison of learners' L1 of the training model Table 5.2 shows the results trained with different learners' L1 languages[7]. Performance was not better when translation model was trained with the same L1 as the test data each method, while perfor-

[7] Note that language model was trained from the whole training corpus. I did not change L1 for language model.

Table 5.1: Comparison of the performance of error correction for each system of different granularity of tokenization

(a) test: L1 English

	W-W	W-W+D	C-C3	C-C5	C-W	C-W+D
R	0.103	0.074	0.099	0.110	**0.142**	0.118
P	0.286	0.269	0.381	**0.404**	0.317	0.285
F	0.150	0.115	0.156	0.172	**0.194**	0.166

(b) test: L1 Mandarin

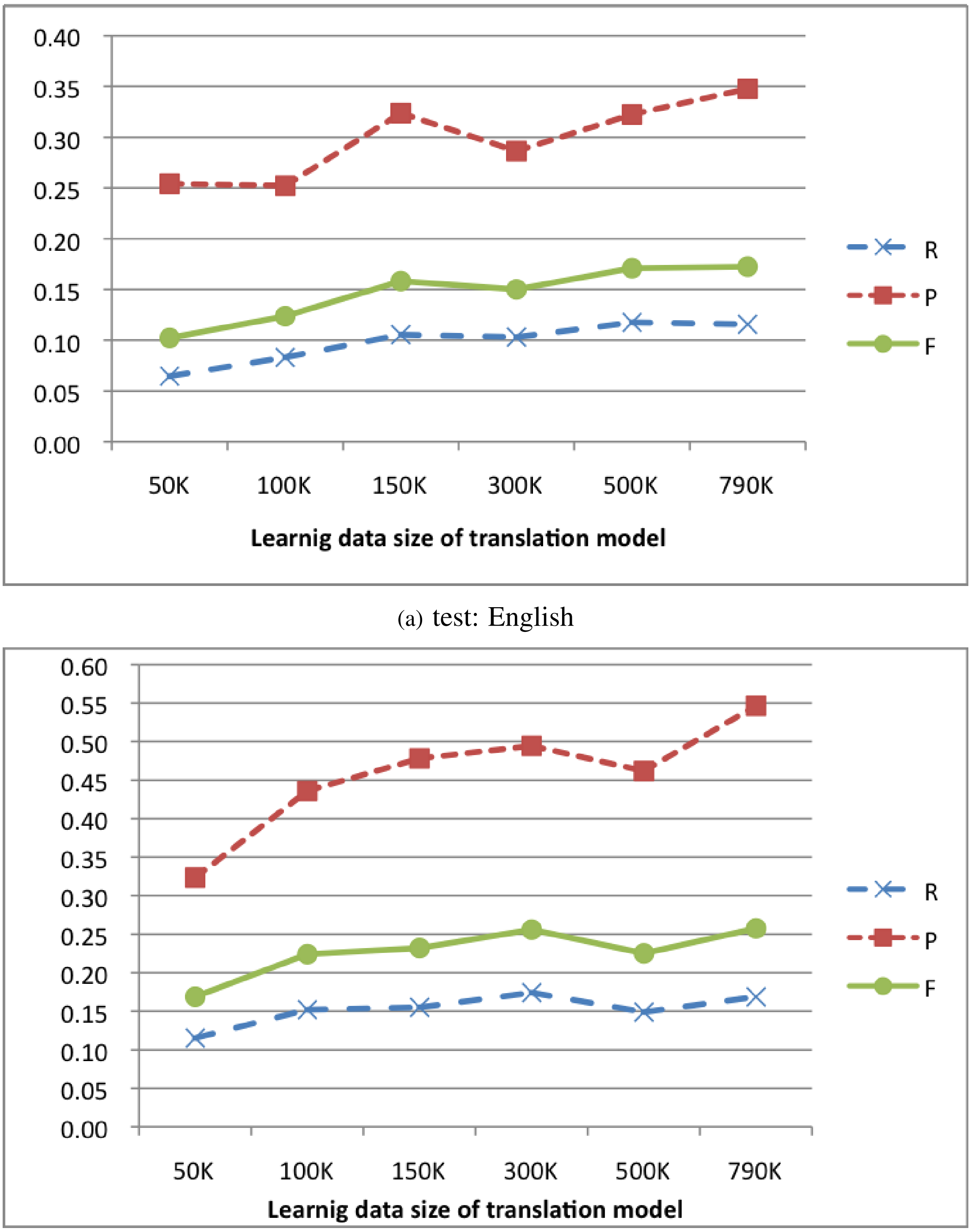
	W-W	W-W+D	C-C3	C-C5	C-W	C-W+D
R	**0.174**	0.152	0.142	0.168	0.139	0.142
P	0.495	**0.537**	0.498	0.494	0.419	0.472
F	**0.256**	0.237	0.219	0.248	0.208	0.217

mance was best when translation model was trained with the same L1 as the test data each method in the whole methods. The variance of results is small for each methods in the case that training data is ALL, while the variance of results is large for each method in the cases that writers' L1 of training data is English or Mandarin. W-W achieved the best F-measure in Figure 5.2(b), because the text written by Mandarin L1 speakers is easy to tokenize.

5.2.3 Experimental Results for NAIST Goyo corpus

I evaluated the performance of error types for NAIST Goyo corpus by hand. Table 5.3 shows recall and number of correct answer using NAIST Goyo corpus as test data. There is not much different in number of correct answer between NAIST Goyo corpus and Lang-8 corpus. However, the system trained by NAIST Goyo coprus is higher performance than Lang-8 about "word selection" error. Recall is low but the error correction systems can correct various types of error. The error type which has the highest recall in all error types is "Spelling" error.

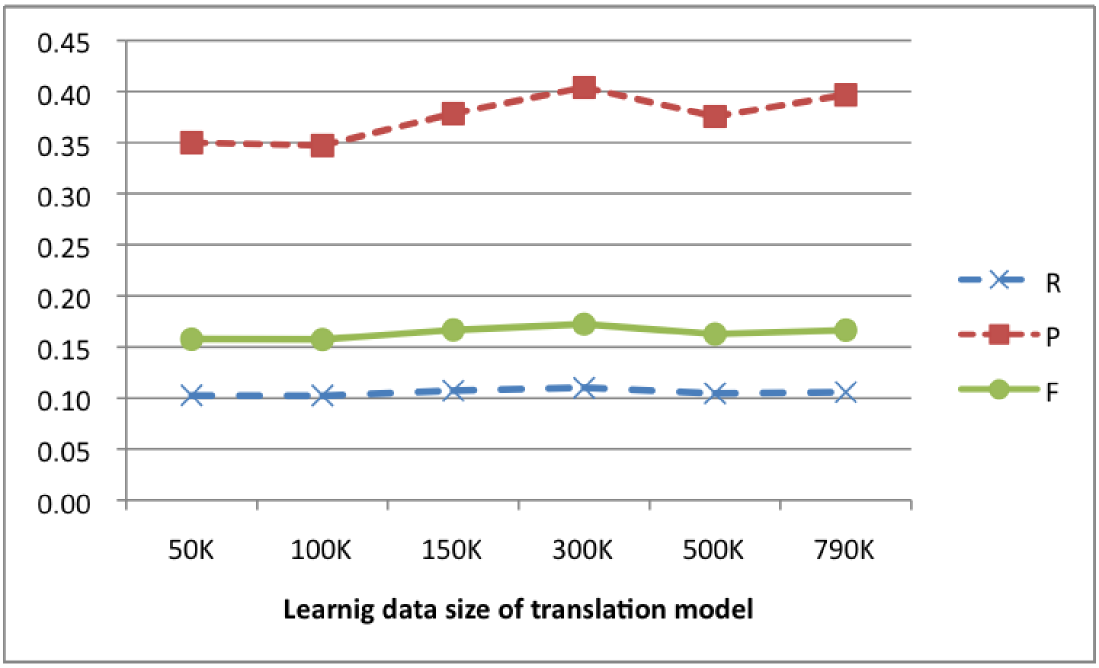
Table 5.4 shows character-based recall, precision and F-score. F-score of W-W model is not different between training corpora. In F-scores of C-C and C-W, using Lang-8 corpus is better than NAIST Goyo corpus.



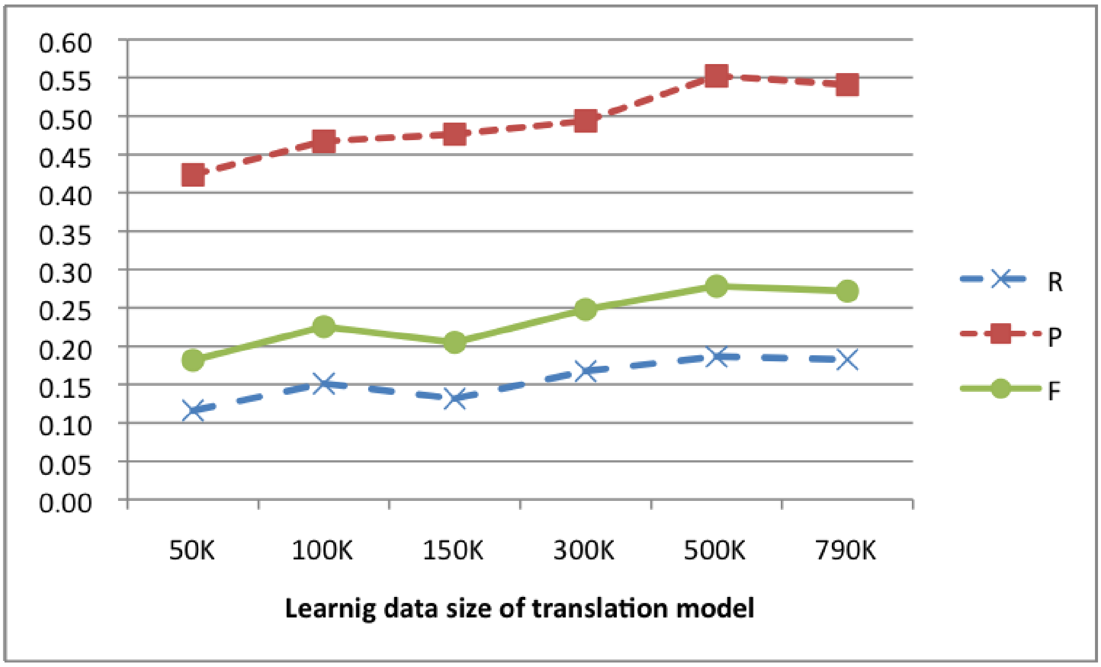
(a) test: English

(b) test: Mandarin

Figure 5.3: Comparison of the performance of error correction for different size of translation model (W-W).

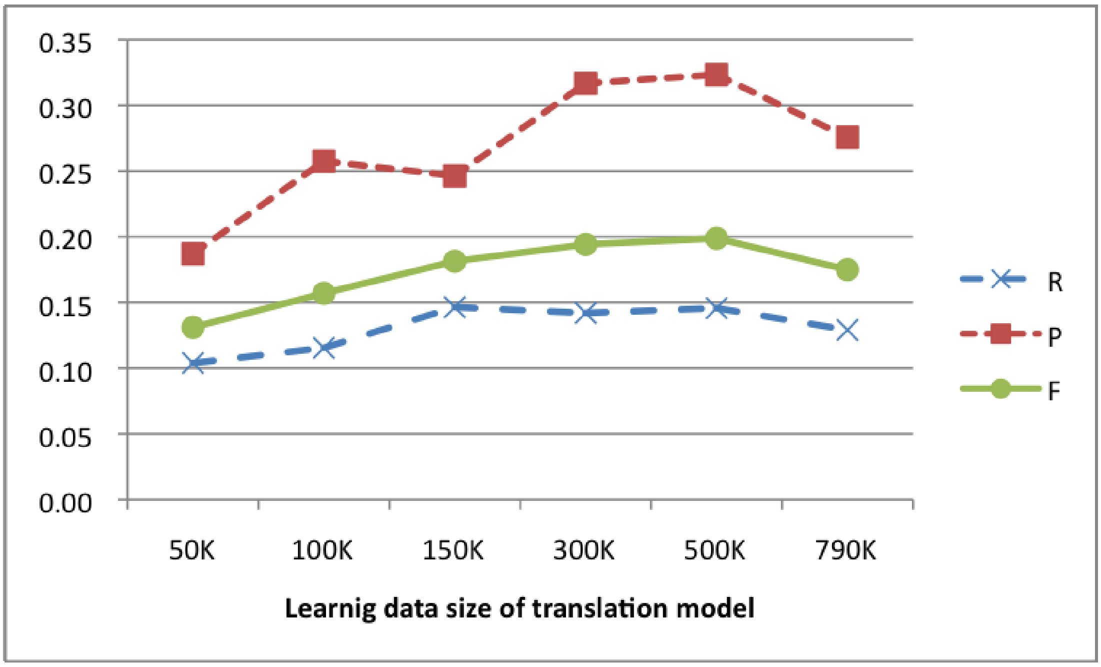


(a) test: English

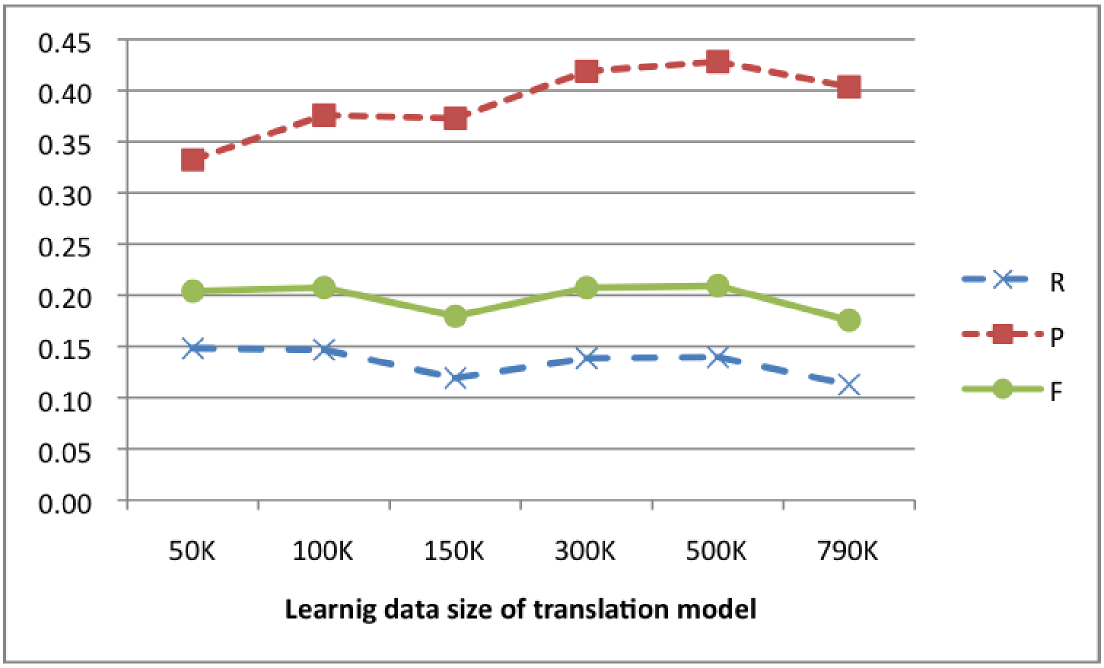


(b) test: Mandarin

Figure 5.4: Comparison of the performance of error correction for different size of translation model (C-C5).



(a) test: English



(b) test: Mandarin

Figure 5.5: Comparison of the performance of error correction for different size of translation model (C-W).

Table 5.2: Comparison of the performance of error correction trained on different first language

(a) test: English

			method		
			W-W	C-C5	C-W
Learners' L1 of training data	English	Recall	0.010	0.128	0.136
		Precision	0.326	0.383	0.271
		F-measure	0.151	**0.190**	0.181
	Mandarin	Recall	0.051	0.089	0.097
		Precision	0.248	0.266	0.189
		F-measure	0.084	0.132	0.126
	ALL	Recall	0.105	0.107	0.146
		Precision	0.324	0.378	0.246
		F-measure	0.158	0.167	0.181

(b) test: Mandarin

			method		
			W-W	C-C5	C-W
Learners' L1 of training data	English	Recall	0.170	0.151	0.171
		Precision	0.457	0.494	0.354
		F-measure	0.245	0.229	0.231
	Mandarin	Recall	0.183	0.116	0.124
		Precision	0.457	0.474	0.389
		F-measure	**0.260**	0.185	0.187
	ALL	Recall	0.155	0.132	0.119
		Precision	0.478	0.476	0.373
		F-measure	0.232	0.205	0.180

Table 5.3: Recall and number of correct answer using NAIST Goyo corpus as test data

Training corpus	NAIST Goyo corpus			Lang-8		
Error type (number of correct answer)	W-W	C-C5	C-W	W-W	C-C5	C-W
Particle (89)	0.056 (5)	0.034 (3)	0.056 (5)	**0.067 (6)**	0.045 (4)	**0.067 (6)**
Word selection (58)	0.069 (4)	0.052 (3)	**0.103 (6)**	0.034 (2)	0.017 (1)	0.017 (1)
Spelling (37)	0.216 (8)	0.243 (9)	0.216 (8)	0.189 (7)	**0.297 (11)**	0.270 (10)
Unnecessary (30)	0.033 (1)	0.000 (0)	**0.067 (2)**	0.000 (0)	0.000 (0)	0.000 (0)
Missing (27)	0.037 (1)	0.000 (0)	0.000 (0)	**0.074 (2)**	0.037 (1)	0.074 (2)
Verb (19)	0.105 (2)	0.105 (2)	0.105 (2)	**0.158 (3)**	**0.158 (3)**	**0.158 (3)**
Style (6)	**0.167 (1)**	0.000 (0)	0.000 (0)	0.000 (0)	0.000 (0)	0.000 (0)
NONE (56)	0.143 (8)	0.107 (6)	**0.161** (9)	0.107 (6)	0.107 (6)	0.143 (8)
合計	0.085 (30)	0.065 (23)	**0.093** (33)	0.073 (26)	0.073 (26)	0.085 (30)

Table 5.4: Character-based recall, precision and F-score using NAIST Goyo corpus as test data

Training corpus	NAIST Goyo corpus			Lang-8		
	W-W	C-C5	C-W	W-W	C-C5	C-W
Recall	0.080	0.048	0.112	0.081	0.072	**0.130**
Precision	0.222	0.231	0.131	0.190	**0.295**	0.236
F-score	0.118	0.080	0.121	0.113	0.116	**0.167**

5.3 Discussion

As I discussed in Chapter 2, the extracted corpus still contains comments in the corrected sentences. However, it does not greatly affect the performance of the JSL learner's error correction, demonstrating that I was able to build a large-scale JSL learners' corpus from revision logs. Moreover, I have checked all the output of my SMT-based error correction system, but none of the errors of the system are derived from the annotators' comments.

Here are some examples illustrating the difference of the scale of the training corpus. I compared translation models trained on 100K sentences and 300K sentences. Note that the model trained on 100K sentences gave the worst result, wheares model trained on 300M sentences achieved the best in Figure 5.4(a). In both cases, the language models were trained on the same 790K sentences. Both models corrected the examples below:

Original: またどもう ありがとう
(Thanks, Matadomou (OOV))

Correct: またどうも ありがとう
(Thank you again)

Also, both of them corrected a case marker error frequently found in JSL learners' writing as in:

Original：TRUTHわ美しいです
(TRUTH wa beautiful)

Correct：TRUTHは美しいです
(TRUTH is beatiful)

On the other hand, the model trained on 300K sentences corrected the following example:

> Original: 学生な る たら学校に行ける
> (the learner made an error in conjugation form.)
>
> Correct: 学生な っ たら学校に行ける
> (Becoming a student, I can go to school.)
>
> 100K: 学生な るため 学校に行ける
> (I can go to school to be student)
>
> 300K: 学生な っ たら学校に行ける
> (Becoming a student, I go to a school)

This example also illustrates the fact that there remains uncorrected errors (missing "ni" case marker after "学生" *student*) as I discussed in Section 2.5.

Here are some examples illustrating the difference of the methods. I compared C-C5 and C-W. Note that both C-C5 and C-W achieve better F-measure than rest of methods in Figure 5.1(a) C-C5 corrected the following example:

> Original: いつも英語 けだ を話したくない
> (I do not want to speak English keda)
>
> Correct: いつも英語 だけ を話したくない
> (I do not want to speak only English)
>
> C-C5: いつも英語 だけ を話したくない
> (I do not want to speak only English)
>
> C-W: いつも英語 だ を話したくない
> (I do not want to speak Englsh da)

On the other hand, C-W corrected the examples below:

> Original: 協会に行 きます つもりです
> (I will going to go an accademy)
>
> Correct: 協会に行 く つもりです
> (I am going to go to an academy)
>
> C-C5: 協会に行 きます つもりです
> (I will going to go an accademy)
>
> C-W: 協会に行 く つもりです
> (I am going to go to an academy)

The word 3gram language model produces an effect. The word 3gram uses wider information than the character 5gram.

Seeing Figures 5.3, 5.4, 5.5, and Table 5.2, the experimental result varies widely by test data or methods. From this, there are effective sentences of training data for the each method and each test data.

The character-based models are better than word-based models for test data whose writers' L1 is English while word-based models are better than character-based models for test data whose writers' L1 is Mandarin. This is because learners whose L1 is Mandarin can write Chinese character (kanji) so that word segmentation is less likely to fail for sentences whose writers' L1 is Mandarin.

Chapter 6

Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings

6.1 Background

English as a Second Language (ESL) learners' writings contain various kinds of grammatical errors. Recent growth in corpus annotation of learner English allows detailed analysis of grammatical errors in learners' writings. Konan-JIEM Learner Corpus (hereafter referred to as KJ Corpus)[1] is one such corpus composed of English essays written by Japanese college students. Table 6.1 shows the distribution of errors found in KJ Corpus[2]. The most frequent error type is *article* errors, followed by *noun number* and *preposition* errors. It is not surprising that frequent types of errors account for the most errors, but it should be noted that there are many different types of errors in learner corpus.

Recently, Swanson and Yamangil (2012) presented a detailed analysis on correcting all types of errors in the Cambridge Learner Corpus, but their task is different from the others in that their goal is to detect errors and select error types given both the original and corrected text, which is not often available in practice.

Some types of errors like agreement errors can be corrected by simple rules using heuristics, while others like preposition errors are difficult to correct without statistical models trained on native corpora and/or learner corpora. It was not until recently that

[1] `http://www.gsk.or.jp/catalog/GSK2012-A/catalog_e.html`

[2] Spelling errors are excluded from target of annotation in KJ Corpus.

Types	**Proportion** (%)	**Types**	**Proportion** (%)
article	19.23	verb other	4.09
noun number	13.88	adverb	3.59
preposition	13.56	conjunction	2.04
tense	8.77	word order	1.34
lexical choice of noun	7.04	noun other	1.30
lexical choice of verb	6.90	auxiliary verb	0.88
pronoun	6.62	other lexical choice	0.74
agreement	5.25	relative	0.42
adjective	4.30	interrogative	0.04

Table 6.1: The distribution of errors on KJ Corpus.

large scale learner corpora became widely available for grammatical error correction. However, little is known about the effect of learner corpus size in ESL grammatical error correction.

In this chapter, I conduct experiments in error correction targeting all types of errors using a large scale error-annotated learner corpus to see the effect of corpus size in grammatical error correction. I build an error correction system with phrase-based statistical machine translation (SMT) technique. Also, I create a large scale error-tagged corpus of learner English from the web. I then analyze the results of error correction by breaking down the error types and discuss the strength and weakness of the example based approach using a large scale but noisy learner corpus.

The main contribution of this work is two-fold:

- To my knowledge, it is the first attempt to use a large scale learner corpus to correct all types of errors.
- I show the effect of learner corpus size on the phrase-based SMT approach and show its advantages and disadvantages.

6.2 Related Work

Even though there are many works on error correction in learners' English, only a few target multiple various kinds of grammatical errors.

First, Brockett et al. (2006) proposed an error correction model with phrase-based SMT. Even though their model can deal with all types of errors, they evaluated their method only on noun number errors using an artificial data, partly because there was no large scale learner corpus available at the time. I would like to emphasize that my work is the first attempt to use a real world large learner corpus with phrase-based SMT technique. I will show that phrase-based SMT especially suffers from data sparseness.

Second, Park and Levy (2011) attempted to correct various kinds of errors with a noisy channel model using a large scale unannotated corpus of learner English. Mine differs from their work in that I use a large scale error-tagged corpus annotated by the wisdom of crowds. In addition, they targeted only spelling, article, preposition and word form errors, while I do not restrict error types.

Third, Han et al. (2010) developed a preposition correction system using a large scale error-tagged corpus of learner English. They built a maximum entropy-based model for preposition errors trained on learner and native corpora. I also take advantage of a large scale error-tagged corpus of learner English, but use phrase-based SMT to deal with various kinds of errors and to fully exploit the learner corpus.

Recently, Dahlmeier and Ng (2012a) presented a beam-search decoder for correcting spelling, article, preposition, punctuation and noun number errors. They reported that their discriminative model achieves considerably better results than an SMT baseline trained on a few hundreds of sentences. As I will see later, I observed a similar tendency in preposition error correction when I trained a phrase-based SMT system on a small learner corpus. However, in this work, I exploit a large scale error-annotated corpus extracted from the web to overcome the data sparseness problem.

6.3 Experiment: Effect of Learner Corpus Size in Grammatical Error Correction

I carried out an experiment on grammatical error correction with SMT-based system using a large scale learner corpus. To see the effect of corpus size, I compare a system using Lang-8 Corpus (large scale learner corpus) with different sizes and a system using KJ Corpus (small scale corpus). In order to get a closer look at the effect of error correction methods, I also experimented on the preposition error correction task using a maximum entropy model as a discriminative baseline and SMT-based models as my proposal for all error correction.

6.3.1 Tools for Statistical Machine Translation

I used Moses 2010-08-13[3] with default parameters as a decoder and GIZA++ 1.0.5[4] as an alignment tool to implement an error correction system with phrase-based SMT. I applied grow-diag-final-and (Och and Ney, 2003) heuristics for phrase extraction. The number of extracted phrases are 1,050,070 (245 MB) using all data of Lang-8 Corpus. I used 3-gram as a language model trained on the corrected text of Lang-8 Corpus.

Next, I built the maximum entropy model (Berger et al., 1996) as a multi-class classifier baseline for preposition error correction (Sakaguchi et al., 2012). I used the implementation of Maximum Entropy Modeling Toolkit[5] with its default parameters. I incorporated surface, POS, WordNet, parse and language model features described in (Tetreault et al., 2010) and (De Felice and Pulman, 2008). POS and parse features were extracted using the Stanford Parser 2.0.2. This system achieves recall of 18.44, precision of 34.88 and F-measure of 24.12 trained and tested on the CLC FCE dataset (Yannakoudakis et al., 2011), which ranked the 4th out of 13 systems at the HOO 2012 Shared Task (Dale et al., 2012).

6.3.2 Experimental Data

I use metadata of users to determine the L1 of English learners. Because my test corpus (KJ Corpus) is written by Japanese college students, I would like to use the same kind of data; it is out side of the scope of this paper to see the effect of learners' L1. There are 509,116 sentence pairs in English writings written by Japanese L1 English learners. However, I need to filter noisy sentences because it may be hard to align them if the sentences are drastically changed from the original learner's sentences, resulting in degraded performance on phrase-based SMT approach. Therefore, I calculate the edit distance between a learner sentence and the corrected sentence using a dynamic programming algorithm, and retain sentences whose numbers of both insertions and deletions is equal to or less than 5 words[6]. As a result, I obtain 391,699 sentence pairs.

I use KJ Corpus as a test data. KJ Corpus consist of 170 essays, containing 2,411 sentences. When I experiment on a system using KJ Corpus, I perform 5-fold cross

[3] `http://http://www.statmt.org/moses/`

[4] `http://code.google.com/p/giza-pp/`

[5] `https://github.com/lzhang10/maxent`

[6] I use 6 as a distortion-limit for Moses, therefore I chose the edit distance to be smaller than the distortion-limit.

validation.

6.3.3 Experimental Results

Table 6.2 shows error correction results for each type of errors on different corpora. I compared SMT systems trained on KJ Corpus, Lang-8 Corpus with the same amount of data with KJ Corpus, and full Lang-8 Corpus. With very few exceptions, the larger the size of learner corpus, the higher the accuracy. In addition, using the larger corpus, precision tends to increase more than recall.

Table 6.3 presents F-measures for each type of error varying the corpus sizes (2K, 10K, 20K, 100K, 200K, 300K, All (390K)). As I will see later in Section 6.3.4, there are two types of errors in which learner corpus size matters.

Table 6.4 shows the performance of preposition error correction. Perhaps not surprising, but it still deserves attention that SMT model trained on all Lang-8 Corpus clearly outperformed other two systems. MaxEnt does slightly better than SMT when they are trained on the same small corpus. Unfortunately, I were not able to use Lang-8 Corpus since it took too long to train.

6.3.4 Discussion

I can classify errors into two types: (1) errors which get better correction by increasing corpus size and (2) errors which have little relationship with corpus size. The first type of errors includes *article*, *preposition*, *lexical choice of noun*, *lexical choice of verb*, *adjective*, and *noun other*. On the other hand, the second type of errors comprises *noun number*, *tense*, *agreement*, *adverb*, *conjunction*, *word order*, *auxiliary verb*, *relative* and *interrogative*. I can expect to improve performance (both recall and precision) for errors that require wide coverage lexical knowledge, such as lexical choice errors, by using a much larger corpus with phrase-based SMT. In contrast, I may say that errors which involve larger context such as tense errors are difficult to correct with phrase-based SMT. I discuss the result while looking at examples of two of the former type of errors (*article* and *lexical choice of noun*) whose F-measures improve with increasing corpus size, and three of the latter type of errors (*noun number*, *tense* and *agreement*), whose F-measures do not change or even degrade.

Table 6.5 shows examples of article and lexical choice of noun. These are the examples that phrase-based SMT failed to correct using KJ Corpus. Because I can acquire a lot of pairs of an error phrase and its correction by increasing the size of the learner corpus, the phrase-based SMT was able to correct them using Lang-8 Corpus.

Table 6.6 shows examples of noun number, tense and agreement errors. The first example of noun number was corrected using Lang-8 Corpus with phrase-based SMT since the error is one of the common learners' expressions. The second was not corrected using Lang-8 Corpus with phrase-based SMT because "dools"[7] is slightly displaced from "a big", and a proper noun "snoopy" is inserted between "dools" and "a big". It is hard to correct this kind of error with Phrase-based SMT, even using artificial data such as in (Brockett et al., 2006). To solve this problem, I need to conduct generalization using POS or consider dependency relations.

The first example of a tense error was corrected using both KJ Corpus and Lang-8 Corpus with phrase-based SMT. One of the reasons why the baseline system was able to correct the error is that it requires only local context to correct and is very frequent even in a small leaner corpus. In the second example, the system fails to find tense agreement in the complex sentence. Tense error is difficult to correct for phrase-based SMT since it involves global context (Tajiri et al., 2012).

The first example of agreement error was corrected using Lang-8 Corpus with phrase-based SMT. This is because the phrase pair correcting "Flowers is" to "Flowers are" is frequent and the language model probability of "Flowers are" is also higher than "Flowers is". The second example is one that the system failed to correct since the pattern is unseen in the learner corpus and thus the system has no way to capture the relation between the subject "reading" and "are". To solve this problem, it needs to get the subject-verb relation considering a dependency structure.

As for prepostion error correction, I suspect that there are two reasons why the SMT-based model using full Lang-8 Corpus outperformed the MaxEnt model. First, due to the small amount of training data in KJ Corpus (2,000 sentences), the MaxEnt model failed to build a high performance system. Second, the high performance of the SMT system may be attributed to the fact that both KJ Corpus and Lang-8 Corpus were written by Japanese native speakers. Also, the reason why the MaxEnt model achieved better result than SMT when trained on the same small corpus is possibly because KJ Corpus is too small to learn variations in learner English by phrase-based SMT approach, while a discriminative model can exploit a small dataset using rich features.

[7] The word "dools" written by a learner is also a spelling error.

Training Corpus	**KJ Corpus**			**Lang-8 Corpus (2K)**			**Lang-8 Corpus (390K)**		
	Recall	Prec	F	Recall	Prec	F	Recall	Prec	F
article	0.187	0.531	0.277	0.187	0.571	0.282	**0.359**	**0.761**	**0.488**
noun number	0.207	0.603	0.308	0.136	0.671	0.226	0.199	**0.710**	0.311
preposition	0.137	0.375	0.201	0.092	0.319	0.143	**0.262**	**0.585**	**0.361**
tense	0.102	0.170	0.128	0.043	0.088	0.058	0.080	0.149	0.104
lexical choice of noun	0.035	0.114	0.054	0.033	0.152	0.054	**0.182**	**0.443**	**0.258**
lexical choice of verb	0.070	0.161	0.098	0.065	0.200	0.098	**0.192**	**0.324**	**0.241**
pronoun	0.075	0.220	0.112	0.040	0.143	0.063	0.150	**0.367**	**0.213**
agreement	0.236	**0.604**	0.340	0.125	0.483	0.199	0.228	0.469	0.307
adjective	0.151	0.326	0.206	0.056	0.286	0.094	**0.389**	**0.522**	**0.446**
verb other	0.089	0.139	0.109	0.147	0.333	0.204	**0.286**	**0.419**	**0.340**
adverb	0.265	0.450	0.333	0.214	0.429	0.286	0.292	0.432	0.349
conjunction	0.100	0.417	0.161	0.091	0.714	0.161	0.115	**0.546**	0.190
word order	0.500	0.025	0.048	0.667	0.050	0.093	**0.750**	0.075	0.136
noun other	0.182	0.222	0.200	0.143	0.167	0.154	**0.571**	**0.429**	**0.490**
auxiliary verb	0.056	0.167	0.083	0.100	0.400	0.160	0.100	**0.400**	0.160
other lexical choice	0.167	0.200	0.182	0.000	0.000	0.000	**0.357**	**0.455**	**0.400**
relative	0.111	0.250	0.154	0.182	0.667	0.286	0.091	**0.500**	0.154
interrogative	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Total	0.149	0.147	0.148	0.113	0.205	0.146	0.247	**0.275**	**0.260**

Table 6.2: Result for each type of errors by statistical machine translation. Bold face indicates that one system's result is equal or greater by more than 0.1 points than the other systems' result.

Training Corpus	KJ	Lang-8						
		2K	10K	20K	100K	200K	300K	390K
article	0.277	0.282	*0.390	*0.420	*0.443	*0.459	*0.475	*0.488
noun number	0.308	0.226	0.214	0.238	0.270	0.300	0.319	0.311
preposition	0.201	0.143	0.192	0.226	*0.333	*0.336	*0.344	*0.362
tense	0.128	0.058	0.066	0.058	0.081	0.096	0.089	0.104
lexical choice of noun	0.054	0.054	0.124	0.133	*0.189	*0.216	*0.250	*0.258
lexical choice of verb	0.098	0.098	0.087	0.138	*0.196	*0.232	*0.232	*0.241
pronoun	0.112	0.063	0.131	0.150	0.177	0.195	0.213	0.213
agreement	0.340	0.197	0.224	0.248	0.260	0.284	0.307	0.307
adjective	0.206	0.094	0.165	0.219	*0.413	*0.426	*0.426	*0.446
verb other	0.109	0.204	0.240	0.311	0.291	*0.340	0.308	0.340
adverb	0.333	0.286	0.286	0.302	0.333	0.349	0.349	0.349
conjunction	0.161	0.161	0.161	0.191	0.161	0.191	0.191	0.191
word order	0.048	0.093	0.093	0.091	0.091	0.091	0.091	0.136
noun other	0.200	0.154	0.286	0.286	*0.531	*0.490	*0.490	*0.490
auxiliary verb	0.083	0.160	0.160	0.083	0.083	0.160	0.160	0.160
other lexical choice	0.182	0.000	0.095	0.095	0.400	0.400	0.400	0.400
relative	0.154	0.285	0.154	0.154	0.154	0.154	0.154	0.154
interrogative	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Total	0.148	0.146	0.180	0.200	0.239	0.247	0.254	0.260

Table 6.3: Results (F-measure) for error correction by SMT varying the learner corpus sizes. Asterisks indicate that the difference of result using Lang-8 Corpus and result using KJ Corpus is statistically significant ($p < 0.01$).

System	Training corpus	Recall	Precision	F-measure
Maximum entropy-based model	KJ Corpus	0.165	0.407	0.235
Phrase-based SMT	KJ Corpus	0.137	0.375	0.201
Phrase-based SMT	Lang-8 Corpus (390K)	**0.262**	**0.585**	**0.362**

Table 6.4: Result for preposition error correction on KJ Corpus.

	learner	**correct**
article	I like a chocolate very much.	I like _ chocolate very much.
lexical choice of noun	my cycle was injured, but i wasn’t.	my bicycle was damaged, but i wasn’t.

Table 6.5: Examples of system output for article and lexical choice of noun error

	learner	**correct**
noun number 1	I read various type books.	I read various types of books.
noun number 2	There is a big snoopy dools in my room.	There is a big snoopy doll in my room.
tense 1	If I ’ll live in saitama, I must have ...	If I _ live in saitama, I must have ...
tense 2	The weather is very sunny, so we were ...	The weather was very sunny, so we were ...
agreement 1	Flowers is very beautiful.	Flowers are very beautiful.
agreement 2	I think, reading comics are not ”reading”	I think, reading comics is not ”reading”

Table 6.6: Examples of system results for noun number, tense and agreement errors. Asterisks indicate that the SMT system using full Lang-8 Corpus failed to correct the errors.

Chapter 7

Grammatical Error Correction Considering Multi-word Expressions

7.1 Background

For dealing with any types of errors, grammatical error correction methods using phrase-based statistical machine translation (SMT) are proposed until previous chapter. Phrase-based SMT carries out translation with phrases which are a sequence of words as translation units. However, since phrases are extracted in an unsupervised manner, an Multi-Word Expression (MWE) like "a lot of" may not be treated as one phrase. In machine translation fields, phrase-based SMT considering MWEs achieved higher performance (Carpuat and Diab, 2010; Ren et al., 2009).

In this chapter, I propose a grammatical error correction method considering MWEs. To be precise, I apply machine translation methods considering MWEs (Carpuat and Diab, 2010) to grammatical error correction. They turn MWEs into single units in the source side sentences (English). Unlike typical machine translation that translates between two languages, in the grammatical error correction task, source side sentences contain errors. Thus, I propose two methods; one is that MWEs are treated as one word in both source and target side sentences, the other is that MWEs are treated as one word in only the target side sentences.

7.2 Related Work

Research on grammatical error correction has recently become very popular. Grammatical error correction methods are roughly divided into two types; (1) targeting few restricted types of errors (Rozovskaya and Roth, 2011; Rozovskaya and Roth, 2013; Tajiri et al., 2012) and (2) targeting any types of errors. In the first type of error correction, classifiers like Support Vector Machines have mainly been used. In the second type, statistical machine translation methods have been used. The only features for grammatical error correction that have been considered in many of previous works are token, POS and syntactic information of single words, and features considering two (or more) words as a whole such as MWEs have never been used.

There is the work dealing with collocations, a kind of MWEs, as target of error detection (Futagi et al., 2008). Our method is different in that we are aiming at correcting not MWEs but other expressions like articles, prepositions and noun numbers as targets considering MWEs.

A lot of research for identifying MWEs and constructing MWE resources have been conducted (Schneider et al., 2014; Shigeto et al., 2013). In addition, there is some research in natural language processing applications using MWEs; i.e., statistical machine translation (Carpuat and Diab, 2010; Ren et al., 2009), information retrieval (Newman et al., 2012) and opinion mining (Berend, 2011).

My task is very similar to the research of SMT using MWEs (Carpuat and Diab, 2010; Ren et al., 2009). However I am in different situation where incorrect words may be included in source sentence side, thus identifying MWEs in source side may make mistakes.

7.3 Multi-word Expressions

MWEs are defined as expressions having "idiosyncratic interpretations that cross word boundaries (or spaces)" (Sag et al., 2002). In this chapter, I mainly deal with fixed expressions that function either as adverbs, conjunctions, determiners, prepositions, prepositional phrases or pronouns.

Table 7.1: The rate of overlap of multi-word expressions from Penn Treebank section of OntoNotes and Lang-8 Learner Corpora

top number	rate of overlap
10	30.0%
20	45.0%
30	46.7%
40	57.5%
50	54.0%
70	57.1%
120	66.7%
170	66.5%

7.3.1 Multi-word Expressions in Native Corpora and Learner Corpora

ESL learners also use a lot of MWEs in their writings just like native speakers. For comparing MWEs usages of ESL learners and native speakers, I prepare a native corpus and a learner corpus. I use the MWE data set from (Shigeto et al., 2013), MWE-annotated Penn Treebank sections of OntoNotes Release 4.0[1] as the native corpus. I use Lang-8 Learner Corpora[2] as the learner corpus[3].

Table 7.1 shows the rate of overlap of multi-word expressions from Penn Treebank section of OntoNotes and Lang-8 Learner Corpora in taking top *N*. Although they are in different domains, MWEs used by learners overlap about 60% with those used by native speakers.

The occurrence frequency of MWEs obeys the Zipf's law. In the learner corpus, top 70 MWEs cover about 50%, top 120 MWEs cover about 80% and top 170 MWEs cover 90% of all the MWEs in the corpus by token count.

[1] `https://catalog.ldc.upenn.edu/LDC2011T03`

[2] `http://cl.naist.jp/nldata/lang-8/`

[3] MWEs are automatically tagged by tools which explained in 7.5.1.

7.3.2 Advantage of Using Multi-word Expressions for Grammatical Error Correction

There are two advantages to use MWEs in grammatical error correction. The first advantage is that it prevents translation of correct parts of MWEs to other words. To illustrate this, let us consider the following example:

> He ate sweets, for example ice and cake.

This sentence does not have grammatical errors, thus error correction systems does not need to correct it. However, the system might correct the word “example”, into the following:

> He ate sweets, for examples ice and cake.

This is because the system has no knowledge of MWEs.

The second advantage is that the system becomes capable of considering longer contexts when using MWEs. To illustrate this, let us consider the following example:

> I have a lot of red apple.

Without considering MWEs, the system takes “I have a”, “have a lot”, “a lot of”, “lot of red”, “of red apple”as word 3-grams, unable to consider the relationship between “a lot of” and “apple”.

7.4 Grammatical Error Correction Methods Using Multi-word Expressions

In this section, I describe my error correction method with MWEs. I use statistical machine translation approaches for grammatical error correction. I apply MWEs to the phrase-based SMT.

I propose two methods for grammatical error correction considering MWEs. Previous research of machine translation using MWEs (Carpuat and Diab, 2010) handled

MWEs in source side sentences by simply turning MWEs into single units (by conjoining the constituent words with underscores). I essentially apply their method to grammatical error correction; however, in my case identifying MWEs might fail because source side sentences contain grammatical errors. Therefore, I propose and compare the following two methods.

Using MWEs in both source side and target side In this method, MWEs are considered in both source side and target side. I show an example in the following:

Source: I have a_lot_of pen.
Target: I have a_lot_of pens.

Using MWEs in target side In this method, MWEs are considered only in target side. I show an example in the following:

Source: I have a lot of pen.
Target: I have a_lot_of pens.

I train both language model and translation model using texts of considering MWEs.

7.5 Experiments of Grammatical Error Correction Using Multi-word Expressions

7.5.1 Experimental Settings

I used cicada 0.3.0[4] for the machine translation tool. This includes a decoder and a word aligner. As the language modeling tool I used expgram 0.2.0[5]. I used ZMERT[6] as the parameter tuning tool.

For automatic identifying MWEs, I use AMALGr 1.0[7] (Schneider et al., 2014). The MWE identification tool is re-trained using the MWE data set tagged by Shigeto et al.

[4] `http://www2.nict.go.jp/univ-com/multi_trans/cicada/`

[5] `http://www2.nict.go.jp/univ-com/multi_trans/expgram/`

[6] `http://cs.jhu.edu/~ozaidan/zmert/`

[7] `https://github.com/nschneid/pysupersensetagger`

Table 7.2: Results of grammatical error correction

		Precision	Recall	F-Score
Baseline (without MWEs)		0.301	0.329	0.314
Source: with MWEs, Target: with MWEs	70 (50%)	0.273	0.378	0.317
	120 (80%)	0.300	0.349	0.322
	170 (90%)	0.279	0.382	0.323
	All	0.292	0.328	0.309
Source: without MWEs, Target: with MWEs	70 (50%)	0.301	0.351	0.324
	120 (80%)	0.293	0.369	0.327
	170 (90%)	0.298	0.367	**0.329**
	All	0.313	0.294	0.304

Table 7.3: Examples of system outputs

Leaner	Last month, she gave me a lot of rice and onion.
Baseline	Last month, she gave me a lot of rice and onion.
with MWE	Last month, She gave me a lot of rice and onions.

(2013) on the Penn Treebank sections of OntoNotes Release 4.0. This is because their annotation was more convenient for my purpose.

The translation model was trained on the Lang-8 Learner Corpora v2.0. I extracted English essays which were written by ESL learners whose native language is Japanese from the corpora and cleaned the noise with the method proposed in Chapter 2. As the results, I got 629,787 sentence pairs. I used a 5-gram language model built on corrected sentences of the learner corpora. Konan-JIEM Learner Corpus are used for evaluation and development data. I use 2,411 sentences for evaluation, and 300 sentences for development.

7.5.2 Experimental Results

As evaluation metrics, I use precision, recall and F-score. I compare phrase-based SMT without using MWEs (baseline) with the two methods explained in Section 7.4. In addition, I varied the number of MWEs used for training the translation model and the language model. This is because MWEs that appear few times may introduce noises. I use top 70 (50%), 120 (80%) and 170 (90%) MWEs described in 7.3.1.

Table 7.2 shows the experimental results. The methods considering MWEs achieved higher F-score than baseline except for the case that uses all MWEs. In addition, using more MWEs inceases the F-score.

7.5.3 Discussion

Using all MWEs shows worse results because infrequent MWEs become noise in training and testing. I got better results when I use MWEs only in the target side. This is likely because learners tend to fail to write MWEs correctly, only writing them in partial forms. One cause of deterioration of precision is that a single word like "many" is wrongly corrected into an MWE like "a lot of", although it is actually not incorrect.

There are two reasons why the performance improved considering MWEs. The first reason is that the system becomes capable of considering the relationship between MWEs which are made up of a sequence of two or more lexemes and words lie adjacent to MWEs. I show an example of system results in Table 7.3. Although the baseline system did not correct the example, the system considering MWEs was able to correct this error. This is because the system was able to consider the MWE "a lot of".

The second reason is that the probabilities of translation model and language model are improved by handling MWEs as single units. Let me consider the two sentences, "There are a lot of pens" and "There is a pen." as examples of language model. Without considering MWEs, the word 3-grams, "There are a" and "There is a", have high probability. With considering MWEs, however, the former trigram becomes to "There a_lot_of pens" and then the probabilities of trigrams that should not be given high probability like "There are a" come to low. The correction performance of articles and prepositions that are likely to become a component word of MWEs is considered to improve by this revision. The number of true positive for article as compared with baseline and MWE (170) of only target side are 190 and 227, respectively. Likewise, the number of true positive for preposition as compared with them are 108 and 121, respectively.

Chapter 8

Reranking for Grammatical Error Correction

8.1 Introduction

SMT systems generate many candidate sentences of translation. The systems score for all candidates and output a sentence which have the highest score as the translation result. However, it is not always true that 1-best result of SMT system is the best output, because the scoring is done only with local features. In other words, SMT N-best (N $>$ 1) results contain better outputs than 1-best result.

There are reranking approaches to solve the scoring problem. Reranking is the method which scores for N-best candidate results of SMT and reorders the results. The advantage of reranking is to calculate the score using global features. Figure 8.1 shows a flow of reranking. First, N-best results are obtained by a grammatical error correction system using SMT for a learner sentence (blue broken line [A] in Figure 8.1). Next, a reranking system re-scores for the N-best results and reorders the results Reranking approaches are proposed for common SMT task (Shen et al., 2004; Carter and Monz, 2011; Li and Khudanpur, 2008; Och et al., 2004). Shen et al. (2004) first use perceptron-like algorithm for reranking of common SMT task. However they used a small number of features. Li and Khudanpur (2008) proposed a reranking approach using large scale discriminative n-gram language model for common SMT task. They extend the method proposed for automatic speech recognition to SMT task (Roark et al., 2007). Carter and Monz (2011) is similar to (Li and Khudanpur, 2008), but they used syntactic features (i.e. POS, parse tree) for reranking of common SMT task.

The reranking approach is used in grammatical error correction using phrase-based

SMT (Felice et al., 2014). Their method uses only language model score, however in the reranking step, the system can consider not only surface but also syntactic feature (POS and parse tree and so on.) like Carter and Monz (2011) used. The syntactic information is not considered in the phrase-based SMT, thus considering syntactic features in the reranking system can improves correction performance. In this chapter, I apply a discriminative reranking methods to the task of grammatical error correction. While reranking by discriminative n-gram language model (Shen et al., 2004) is not effective for grammatical error correction, reranking using syntactic features improves $F_{0.5}$ score.

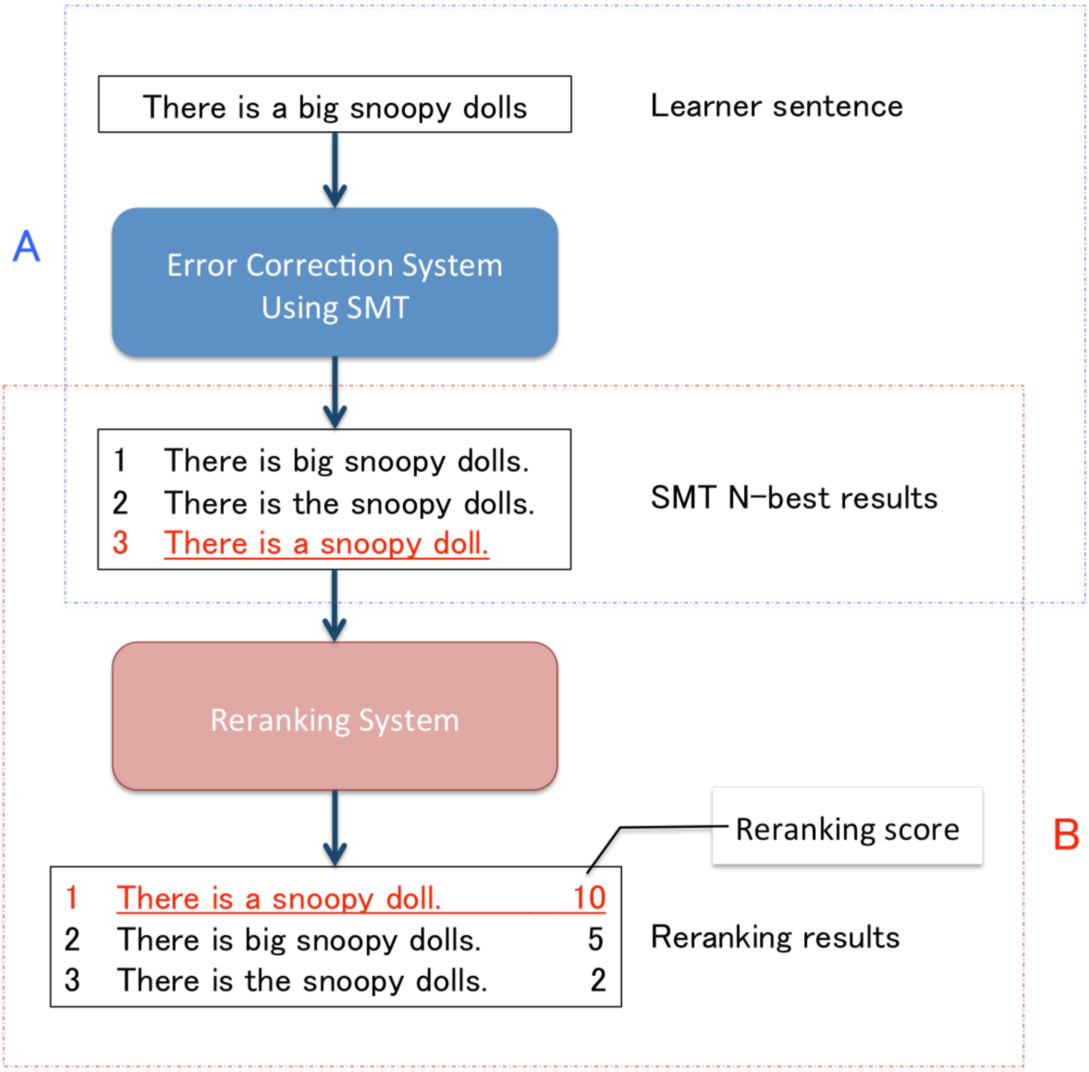

There is a big snoopy dolls
Learner sentence
A
Error Correction System Using SMT
1 There is big snoopy dolls.
2 There is the snoopy dolls.
3 There is a snoopy doll.
SMT N−best results
Reranking System
Reranking score
B
1 There is a snoopy doll. 10
2 There is big snoopy dolls. 5
3 There is the snoopy dolls. 2
Reranking results

Figure 8.1: Flow of reranking

8.2 Why Needs Reranking on Grammtical Error Correction?

The grammatical error correction using SMT has the same problem as common SMT. It is not always true that 1-best correction by the system is the best correction. To prove this, I conducted grammatical error correction experiment using SMT and calculate *n*-best oracle score. The oracle score are calculated by selecting the correction candidates with the highest score from *n*-best lists for each sentence.

Table 8.1 shows oracle scores of baseline grammatical error correction system using SMT[1]. While $F_{0.5}$-Score of 1-best output is 37.9, $F_{0.5}$ of 10-best oracle score is 64.3. The more n-best is increased, the higher oracle score becomes. From this results, 1-best correction by grammatical error correction system using SMT is not always the best correction.

Advantage of Reranking There are three advantages to use the reranking approach for grammatical error correction. First advantage is that the raranking system can use POS, syntactic features which phase-based SMT can not deal with. Some errors need to consider the relation between distant words; i.e., article relation between *a* and *dolls* in *a big Snoopy dolls*.

Second advantage is that POS tagger and parsers can analyze the error-corrected candidate more properly than erroneous sentences, leading to obtain more accurate features. Thus it is promising that taggers for N-best outputs of the system work much better than for learners' original sentences.

Finally, different from pipeline systems (i.e., correcting article errors after correcting noun errors), the reranking system can avoid to correct in conflict. This is because the reranking systems do not correct errors but calculate quality of sentence.

8.3 Proposed Method

[1] See 8.4.1 for a baseline system

Table 8.1: Oracle score of grammatical error correction using SMT

N-best	Precision	Recall	$F_{0.5}$
1	43.9	24.5	37.9
5	71.3	34.0	58.4
10	79.1	36.7	64.3
20	85.0	39.8	69.3
30	87.7	41.4	71.7
40	88.8	42.2	72.7
50	89.5	43.1	73.6
60	90.0	43.7	74.2
70	90.4	44.3	74.9
80	91.3	44.8	75.6
90	91.9	45.0	76.0
100	92.3	45.3	76.4
200	93.6	47.4	78.4
300	94.6	48.3	79.4
400	95.0	49.0	80.0
500	95.2	49.5	80.4
965	96.4	50.6	81.6

Table 8.2: Features for reranking. Examples show features for a sentence *I agree with this statement to a large extent*

Feature name	Examples
Word 2,3-gram	I agree; I agree with; agree with; agree with this; this statement
POS 2,3,4,5-gram	PRP VBP; PRP VBP IN; PRP VBP IN DT; PRP VBP IN DT NN
POS-function word 2,3,4,5-gram	PRP VBP; PRP VBP with; PRP VBP with this; PRP VBP with this NN
Dependency	nsubj(agree, I); det(statement, this); prep_with(agree, statement)

In this section, I explain the discriminative reranking method and the features of the reranker for grammatical error correction.

8.3.1 Discriminative Rearanking Method

In this paper, I use discriminative reranking algorithm using perceptron which successfully exploits syntactic features for n-best reranking for common translation task (Carter and Monz, 2011). Figure 8.2 shows the standard perceptron algorithm for reranking. T is the number of iterations for perceptron learned. N is the number of sentences in training corpus. $GEN(x)$ is n-best list generated by grammatical error correction system using SMT for the input sentence. $ORACLE(x^i)$ determines the best correction for each of the n-best lists according to the $F_{0.5}$. w is weight vector for features and ϕ is feature vector for candidate sentences. Selecting sentence with the highest score from candidate sentences (line 5), if selected sentence and oracle sentence match, then algorithm goes to next sentence. Otherwise, the weight vector is updated.

The disadvantage of perceptron is instability when training data is not linearly separable. As solutions for this problem, an averaged perceptron algorithm was proposed (Freund and Schapire, 1999). In this algorithm, weight vector w_{avg} is defined as:

$$w_{avg} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N} w_t^i \tag{8.1}$$

To select the best correction from n-best candidates, I use the following formula;

$$S(z) = \beta\phi_0(z) + w \cdot \phi(z) \tag{8.2}$$

$\phi_0(z)$ is the score by SMT system for each translation hypothesis. This score is weighted by β. It is possible to use $\phi_0(z)$ as a feature in the perceptron algorithm, but this may lead to under-training (Sutton et al., 2006). I select value for β using development data.

8.3.2 Features of Discriminative Reranker for Grammatical Error Correction

In this paper, I use the features used in (Carter and Monz, 2011) and the features which are used for determiner error correction in (Dahlmeier et al., 2012). In addition,

```
$w \leftarrow 0$
for $t = 1$ to $T$ do
    for $i = 1$ to N do
        $y^i \leftarrow ORACLE(x^i)$
        $z^i \leftarrow argmax_{x \in GEN(x^i)} \phi(z) \cdot w$
        if $z^i \neq y^i$ then
            $w \leftarrow w + \phi(y^i) - \phi(z^i)$
        end if
    end for
end for
return $w$
```

Figure 8.2: Perceptron algorithm for reranking

new simple features of POS and dependency are used. The features in (Carter and Monz, 2011) and my features are extracted from the whole sentence.

I use the features extracted from POS tag sequence, shallow parse tag sequence and shallow parse tag plus POS tag sequence (Carter and Monz, 2011). From these sequence, the features is extracted with the following three definition:

1. $(t_{i-2}t_{i-1}t_i)$, $(t_{i-1}t_i)$, (t_iw_i)
2. $(t_{i-2}t_{i-1}w_i)$
3. $(t_{i-2}w_{i-2}t_{i-1}w_{i-1}t_iw_i)$, $(t_{i-2}t_{i-1}w_{i-1}t_iw_i)$, $(t_{t-1}w_{i-1}t_iw_i)$, $(t_{i-1}t_iw_i)$

Here w_i is a word at position i and t_i is a tag (POS or shallow parse tag) at position i.

The features for determiner error correction are selected from Table 1 in (Dahlmeier et al., 2012). I use lexical features, POS features, head word features and dependency features from (Dahlmeier et al., 2012).

Table 8.2 shows my new features. For POS-function n-gram, I use surface form for words in stop word list, otherwise I use POS tags. I extract “nominal subject (nsubj)”, “determiner (det)”, “direct object (dobj)”, “auxiliary (aux)”, “passive auxiliary (auxpass)”, “numeric modifier (num)” and “prepositional modifier and object of preposition (prep and pobj)” from Stanford Dependency (de Marneffe et al., 2006) for the dependency features.

8.4 Experiments: Reranking for SMT outputs

Table 8.3: Experimental results. TP, FN and FP denote true positive, false negative and false positive, respectively. ArtOrDet and Prep denote “article or determiner” error and “preposition” error, respectively.

		Precision	Recall	$F_{0.5}$	TP	FN	FP	TP of ArtOrDet	TP of Prep
1	1-best result of SMT	43.9	24.5	37.9	598	1847	764	166	56
2	Reranking by n-gram LM	39.5	**31.7**	37.6	834	1797	1280	203	94
3	Ginger	**49.3**	17.1	35.8	398	1932	410	42	33
4	1checker	29.4	17.7	26.0	404	1878	969	76	38
5	CAMB (CoNLL2014)	39.7	30.1	37.3	772	1793	1172	189	101
6	CUUI (CoNLL2014)	41.8	24.9	36.8	623	1881	868	236	36
Discriminative reranking									
7	Word 2,3-gram	43.2	25.4	38.1	644	1838	846	182	63
8	Features of Dahlmeier (2012)	45.2	24.3	38.5	594	1851	721	160	56
9	Features of Carter (2011)	44.9	25.7	39.1	633	1829	776	177	58
10	Our simple features (Table 8.2)	45.0	26.6	**39.5**	662	1825	810	184	66
11	All features (8+9+10)	44.8	26.3	39.3	648	1816	799	177	61

I conducted experiment on grammatical error correction using SMT to see the effect of discriminative reranking.

8.4.1 Experimental Settings

I used phrase-based SMT which many previous research used for grammatical error correction for a baseline system. I used cicada 0.3.5[2] for the phrase-based machine translation tool. This includes a decoder and a word aligner. As the language modeling tool I used KenLM toolkit[3]. I used ZMERT[4] as the parameter tuning tool. I implemented Averaged-perceptron for reranking system.

The translation model was trained on the Lang-8 Learner Corpora v2.0. I extracted English essays which were written by ESL learners and cleaned the noise with the method proposed in (Mizumoto et al., 2011). As the results, I got 1,069,127 sentence pairs. I used a 5-gram language model built on the "Associated Press Worldstream English Service" from English Gigaword corpus and NUCLE 3.2 (Dahlmeier et al., 2013). I used these two language models as separate feature functions in the SMT systems.

For training data of reranking, Lang-8 Learner Corpora is split into 10 parts and each part is corrected by a grammatical error correction system trained on the other 9 parts. I select 10 as *N* for N-best reranking. The reranking system is trained by including gold data that are annotated by native speakers. This is because the system is trained in such a way as to have higher weight on features which appear frequently in correct sentences.

CoNLL-2013 test sets are split into 700 sentences for parameter tuning of SMT and 681 sentences for tuning reranking of parameter beta. CoNLL-2014 test sets, 1,312 sentences are used for evaluation.

I used M2 Scorer as evaluation tool (Dahlmeier and Ng, 2012b). This scorer calculates precision, recall and $F_{0.5}$. I used $F_{0.5}$ as tuning metric.

[2] `http://www2.nict.go.jp/univ-com/multi_trans/cicada/`

[3] `https://kheafield.com/code/kenlm/`

[4] `http://cs.jhu.edu/~ozaidan/zmert/`

8.4.2 Experimental Result and Discussion

Table 8.3 shows the experimental result. I use 1-best result of grammatical error correction system using SMT and reranking by probability of large n-gram language model (Felice et al., 2014) as baseline systems. In addition, I compare the systems which are ranked first and second (Felice et al., 2014; Rozovskaya et al., 2014) in CoNLL2014 Shared Task. I also show the results of English grammaer checker on the Web, Ginger and 1checker.

The discriminative reranking system with my simple features achieved the best $F_{0.5}$ score. Simply adopting large n-gram language model to reranking, recall increases a lot but precision drops. This result is very similar to CAMB system, because CAMB system is SMT-based error correction and reranks using large n-gram language model. Comparing reranking system with my simple feature to CUUI, my system is better in all metrics excluding number of TP of ArtOrDet.

Using the discriminative reranking with my simple features, both precision and recall increase, and number of true positive of both ArtOrDet and Prep increases. Features of Dahlmeier (for article errors) are better $F_{0.5}$ than baseline, however number of true positive decreases. The reranking using all features is less $F_{0.5}$ than using only my simple features. One of this reason is that role of features overlaps.

Chapter 9

Conclusions

I proposed to extract a large-scale learners' corpus from the revision log of a language learning SNS. This corpus is easy to obtain on a large-scale, covers a wide variety of topics and styles, and can be a great source of knowledge for both language learners and instructors. This revision logs also include native language translation by learners own for sentences of learning language. I plan to construct the learner corpus with native language translation from the revision log of a language learning SNS.

I adopted phrase-based SMT approaches to Japanese grammatical error correction task. I proposed the character-wise models to alleviate the problem of erroneous input from language learners. Basically experimental results show that the character-wise models and the combined models of character and word segmentations outperform the word-wise models. My Japanese error correction system still remains some problem. To improve the performance, learning level of writers is predicted, and the grammatical error correction system needs to consider learning level of writers.

For English, I conducted experiments in grammatical error correction targeting all types errors using statistical machine translation technique and I analyze the strength and weakness of grammatical error correction using statistical machine translation. I also proposed two grammatical error correction methods. One is the method considering Multi-word expression. Another is the method using discriminative reranking with POS/syntactic features.

I proposed a straightforward application of MWEs to grammatical error correction, but experimental results show that MWEs have quite good effects on grammatical error correction. Experimental results show that the methods considering MWEs achieved higher F-score than baseline except for the case that uses all MWEs. I plan to use more multi-word expressions which we did not handle in this paper, such as phrasal verbs.

Moreover, I plan to conduct grammatical error correction considering MWEs which contain gaps that are dealt with (Schneider et al., 2014).

I proposed the reranking approach for grammatical error correction with phrase-based SMT. This approach has three advantages; (1) This approach can use global features which phrase-based SMT can not deal with, (2) POS tagger and parsers can analyze the error-corrected candidate, (3) The reranking system can avoid to correct in conflict. My reranking system achieved $F_{0.5}$ score of 39.5 (increasing 1.6 points from the baseline system) on the CoNLL2014 Shared Task test set. I show that POS and dependency features are effective for reranking of grammatical error correction with SMT. I plan to use other features i.e., the features which are used in a classification task, such as preposition correction and tense error correction.

There still remains several topics to explore. First, there are many statistical machine translation methods besides phrase-based SMT I used in this thesis. A factored translation model (Koehn and Hoang, 2007) and a hierarchical phrase-based model (Chiang, 2005) can use more rich information such as hierarchical structures and POS. Moreover adopting a String-to-Tree model (Galley et al., 2006) to grammatical error correction, the error correction system can consider syntactic information. Green and Denero (2012) proposed class-based agreement model for inflected translation. Using this method, grammatical error correction systems may correct inflected error learners make, i.e., subject-verb agreement.

Second, it is not clear how automatic grammatical error correction has positive effect for second language learning. To find out this, I plan to provide my grammatical error correction system using discriminative reranking and collect learners' sentences. I will implement the grammatical correction system to Chantokun[1] which our laboratory has released.

[1] http://cl.naist.jp/chantokun/

Bibliography

[Barzilay and McKeown, 2001] Regina Barzilay and Kathleen R. McKeown. 2001. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 50–57.

[Berend, 2011] Gábor Berend. 2011. Opinion Expression Mining by Exploiting Keyphrase Extraction. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1162–1170.

[Berger et al., 1996] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

[Brockett et al., 2006] Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL Errors Using Phrasal SMT Techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256.

[Brown et al., 1993] 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):266–311.

[Carpuat and Diab, 2010] Marine Carpuat and Mona Diab. 2010. Task-based Evaluation of Multiword Expressions: a Pilot Study in Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 242–245.

[Carter and Monz, 2011] Simon Carter and Christof Monz. 2011. Syntactic Discriminative Language Model Rerankers for Statistical Machine Translation. *Machine Translation*, 25(4):317–339.

[Chiang, 2005] David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270.

[Dahlmeier and Ng, 2012a] Daniel Dahlmeier and Hwee Tou Ng. 2012a. A Beam-Search Decoder for Grammatical Error Correction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 568–578.

[Dahlmeier and Ng, 2012b] Daniel Dahlmeier and Hwee Tou Ng. 2012b. Better Evaluation for Grammatical Error Correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572.

[Dahlmeier et al., 2012] Daniel Dahlmeier, Hwee Tou Ng, and Eric Jun Feng Ng. 2012. NUS at the HOO 2012 Shared Task. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*, pages 216–224.

[Dahlmeier et al., 2013] Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of the 8th Workshop on Building Educational Applications Using NLP*, pages 22–31.

[Dale and Kilgarriff, 2011] Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249.

[Dale et al., 2012] Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A Report on the Preposition and Determiner Error Correction Shared Task. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*, pages 54–62.

[De Felice and Pulman, 2008] Rachele De Felice and Stephen G. Pulman. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 169–176.

[de Marneffe et al., 2006] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase

Structure Trees. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

[Felice et al., 2014] Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24.

[Freund and Schapire, 1999] Yoav Freund and Robert E. Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37(3):277–296.

[Futagi et al., 2008] Yoko Futagi, Paul Deane, Martin Chodorow, and Joel Tetreault. 2008. A Computational Approach to Detecting Collocation Errors in the Writing of Non-Native Speakers of English. *Computer Assisted Language Learning*, 21(4):353–367.

[Galley et al., 2006] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968.

[Green and Denero, 2012] Spence Green and John Denero. 2012. A class-based agreement model for generating accurately inflected translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 146–155.

[Han et al., 2010] Na-Rae Han, Joel Tetreault, Soo-Hwa Lee, and Jin-Young Ha. 2010. Using an Error-Annotated Learner Corpus to Develop an ESL/EFL Error Correction System. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 763–770.

[Imaeda et al., 2003] Koji Imaeda, Atsuo Kawai, Yuji Ishikawa, Ryo Nagata, and Fumito Masui. 2003. Error Detection and Correction of Case particles in Japanese Learner's Composition (in Japanese). In *Proceedings of the Information Processing Society of Japan SIG*, pages 39–46.

[Izumi et al., 2004] Emi Izumi, Kiyotaka Uchimoto, and Hitoshi Isahara. 2004. The NICT JLE Corpus Exploiting the language learners' speech database for research and education. *International Journal of The Computer, the Internet and Management*, 12(2):119–125.

[Koehn and Hoang, 2007] Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876.

[Li and Khudanpur, 2008] Zhifei Li and Sanjeev Khudanpur. 2008. Large-scale discriminative n-gram language models for statistical machine translation. In *Proceedings of the The Association for Machine Translation in the Americas 2008*.

[Liu et al., 2011] Xiaohua Liu, Bo Han, and Min Zhou. 2011. Correcting Verb Selection Errors for ESL with the Perceptron. In *Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 411–423.

[Mizumoto et al., 2011] Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 147–155.

[Nagata et al., 2011] Ryo Nagata, Edward Whittaker, and Vera Sheinman. 2011. Creating a Manually Error-tagged and Shallow-parsed Learner corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1210–1219.

[Nampo et al., 2007] Ryota Nampo, Hokuto Ototake, and Kenji Araki. 2007. Automatic Error Detection and Correction of Japanese Particles Using Features within Bunsetsu (in Japanese). In *Proceedings of the Information Processing Society of Japan SIG*, pages 107–112.

[Newman et al., 2012] David Newman, Nagendra Koilada, Jey Han Lau, and Timothy Baldwin. 2012. Bayesian Text Segmentation for Index Term Identification and Keyphrase Extraction. In *Proceedings of the 24nd International Conference on Computational Linguistics*, pages 2077–2092.

[Ng et al., 2013] Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12.

[Ng et al., 2014] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

[Nicholls, 2003] Diane Nicholls. 2003. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003*, pages 572–581.

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

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

[Och et al., 2004] Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2004 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 161–168.

[Och, 2003] Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

[Oyama and Matsumoto, 2010] Hiromi Oyama and Yuji Matsumoto. 2010. Automatic Error Detection Method for Japanese Case Particles in Japanese Language Learners. In *Proceedings of the Corpus, ICT, and Language Education*, pages 235–24.

[Papineni et al., 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

[Park and Levy, 2011] Y. Albert Park and Roger Levy. 2011. Automated Whole Sentence Grammar Correction Using a Noisy Channel Model. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 934–944.

[Ren et al., 2009] Zhixiang Ren, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang. 2009. Improving Statistical Machine Translation Using Domain Bilingual Multiword Expressions. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 47–54.

[Roark et al., 2007] Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative n-gram language modeling. *Computer Speech Language*, 21(2):373–392.

[Rozovskaya and Roth, 2011] Alla Rozovskaya and Dan Roth. 2011. Algorithm Selection and Model Adaptation for ESL Correction Tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 924–933.

[Rozovskaya and Roth, 2013] Alla Rozovskaya and Dan Roth. 2013. Joint Learning and Inference for Grammatical Error Correction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 791–802.

[Rozovskaya et al., 2014] Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. The Illinois-Columbia System in the CoNLL-2014 Shared Task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 34–42.

[Sag et al., 2002] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15.

[Sakaguchi et al., 2012] Keisuke Sakaguchi, Yuta Hayashibe, Shuhei Kondo, Lis Kanashiro, Tomoya Mizumoto, Mamoru Komachi, and Yuji Matsumoto. 2012. NAIST at the HOO 2012 Shared Task. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*, pages 281–288.

[Schneider et al., 2014] Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A. Smith. 2014. Discriminative Lexical Semantic Segmentation with Gaps: Running the MWE Gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

[Shen et al., 2004] Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative Reranking for Machine Translation. In *Proceedings of Human Language Technologies: The 2004 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

[Shigeto et al., 2013] Yutaro Shigeto, Ai Azuma, Sorami Hisamoto, Shuhei Kondo, Tomoya Kouse, Keisuke Sakaguchi, Akifumi Yoshimoto, Frances Yung, and Yuji Matsumoto. 2013. Construction of English MWE Dictionary and its Application to POS Tagging. In *Proceedings of the 9th Workshop on Multiword Expressions*, pages 139–144.

[Sutton et al., 2006] Charles Sutton, Michael Sindelar, and Andrew McCallum. 2006. Reducing Weight Undertraining in Structured Discriminative Learning. In *Proceedings of Human Language Technologies: The 2006 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 89–95.

[Suzuki and Toutanova, 2006] Hisami Suzuki and Kristina Toutanova. 2006. Learning to Predict Case Markers in Japanese. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1056.

[Swanson and Yamangil, 2012] Ben Swanson and Elif Yamangil. 2012. Correction Detection and Error Type Selection as an ESL Educational Aid. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 357–361.

[Tajiri et al., 2012] Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and Aspect Error Correction for ESL Learners Using Global Context. In

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 198–202.

[Tetreault et al., 2010] Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using Parse Features for Preposition Selection and Error Detection. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 353–358.

[Xue and Hwa, 2010] Huichao Xue and Rebecca Hwa. 2010. Syntax-Driven Machine Translation as a Model of ESL Revision. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1373–1381.

[Yannakoudakis et al., 2011] H. Yannakoudakis, T. Briscoe, and B. Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189.

業績リスト

論文誌

- 水本智也, 小町守, 永田昌明 (NTT), 松本裕治. 日本語学習者の作文自動誤り訂正のための語学学習 SNS の添削ログからの知識獲得. 人工知能学会論文誌, Vol.28, No.5, pp.420-432, July 2013.

国際会議（査読あり）

- Tomoya Mizumoto, Masato Mita, Yuji Matsumoto. Grammatical Error Correction Considering Multi-word Expressions. In Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications (NLP-TEA-2), Beijing, China, August 2015.

- Keisuke Sakaguchi, Tomoya Mizumoto, Mamoru Komachi and Yuji Matsumoto. Joint English Spelling Error Correction and POS Tagging for Language Learners Writing. In Proceedings of the 24th International Conference on Computational Linguistics (COLING-2012), pp.2357-2374, Mumbai, India, December 2012.

- Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata (NTT) and Yuji Matsumoto. The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings. In Proceedings of the 24th International Conference on Computational Linguistics (COLING-2012), pp.863-872, Mumbai, India, December 2012.

- Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata (NTT), Yuji Matsumoto. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In Proceedings the 5th International Joint Conference on Natural Language Processing (IJCNLP2011), pp.147-155, Chiang Mai, Thailand, November 2011.

- Ryo Nagata (Konan Univ.), Tomoya Mizumoto, Kotaro Funakoshi (HRI-JP), Mikio Nakano (HRI-JP). Toward a Chanting Robot for Interactively Teaching English to Children. In Proceedings of the INTERSPEECH 2010 Satellite Workshop on Second Language Studies:Acquisition, Learning, Education and Technology (L2WS2010), Tokyo, Japan, September 2010.

国際会議（その他）

- Ippei Yoshimoto, Tomoya Kose, Kensuke Mitsuzawa, Keisuke Sakaguchi, Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Yuji Matsumoto. NAIST at 2013 CoNLL Shared Task Grammatical Error Correction. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pp.26-33, August 2013.

- Tomoya Mizumoto, Yuta Hayashibe, Keisuke Sakaguchi, Mamoru Komachi, Yuji Matsumoto. NAIST at the NLI 2013 Shared Task. In Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications, pp.134-139, Atlanta, America, June 2013.

- Keisuke Sakaguchi, Yuta Hayashibe, Shuhei Kondo, Lis Kanashiro, Tomoya Mizumoto, Mamoru Komachi, Yuji Matsumoto. NAIST at the HOO 2012 Shared Task. In Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications. pp.281-288, Montreal, Canada, June 2012.

- Kotaro Funakoshi (HRI-JP), Tomoya Mizumoto, Ryo Nagata (Konan Univ.), Mikio Nakano (HRI-JP). The Chanty Bear: A New Application for HRI Research. In Proceedings of the International Conference on HUMAN-ROBOT INTERACTION 2011, pp.141-142, Lausanne, Switzerland, March 2011.

- Kotaro Funakoshi (HRI-JP), Tomoya Mizumoto, Ryo Nagata (Konan Univ.), Mikio Nakano (HRI-JP). The Chanty Bear: Toward a Robot Teaching English to Children. In Proceedings of the International Conference on HUMAN-ROBOT INTERACTION 2011 Workshop: Robots with Children: Practices for Human-Robot Symbiosis, Lausanne, Switzerland, March, 2011.

解説記事

- 水本智也, 小町守. 「なんで日本語はこんなに難しいなの？–リアルな日本語学習者コーパスの分析と言語処理の課題–」. 情報処理, Vol.53, No.3, pp.217-223, March 2012.

国内会議等

- 水本智也, 三田 雅人. Project Next 英文校正タスク: 前置詞誤りを対象とした誤り分析. 言語処理学会第 21 回年次大会ワークショップ, March 2015.
- 水本智也, 松本裕治. 統計的機械翻訳を用いた英語文法誤り訂正のリランキングによる性能改善. 情報処理学会 第 77 回全国大会, March 2015.
- 三田 雅人, 水本智也. Project Next 英文校正タスクの前置詞誤りエラー分析に向けて. NLP 若手の会 第 9 回シンポジウム. September 2014.
- 水本智也. 語学学習 SNS の添削ログからの母語訳付き学習者コーパスの構築に向けて. 第 6 回コーパス日本語学ワークショップ予稿集, pp.215-220, September 2014.
- 水本智也, 松本裕治. 複単語表現を考慮した英語文法誤り訂正. 情報処理学会研究報告 自然言語処理研究会報告, Vol. 2014-NL-217, pp.1-4, July 2014.
- 水本智也, 松本裕治. 統計的機械翻訳を用いた英語文法誤り訂正の結果をリランキングすることで訂正性能の改善はできるか？. 情報処理学会研究報告 自然言語処理研究会報告, Vol. 2014-NL-216, pp.1-5, May 2014.
- 水本智也, 松本裕治. 統計的機械翻訳に基づく英語文法誤り訂正におけるフレーズベースと統語ベースの比較と分析. 言語処理学会第 20 回年次大会 発表論文集, pp.258-261, March 2014.
- 三澤賢祐, 酒井啓道, 吉川友也, 水本智也, 松本裕治. 格構造に注目した日本語-日本語手話の並び替えと述語項構造に注目した語義曖昧性解消. 第 27 回人工知能学会全国大会論文集, pp.1-4, June 2013.
- 永田亮 (甲南大学), 水本智也, Edward Whittaker (Inferret Ltd.). 前置詞誤り検出／訂正のための誤り格フレームの生成. 言語処理学会第 19 回年次大会 発表論文集, pp.616-619, March 2013.

- 水本智也, 林部祐太, 小町守, 永田昌明 (NTT), 松本裕治. 大規模英語学習者コーパスを用いた英作文の文法誤り訂正の課題分析. 情報処理学会研究報告 自然言語処理研究会報告, Vol.2012-NL-209, pp.1-8, November 2012.
- 水本智也, 林部祐太, 坂口慶祐, 小町守, 松本裕治. 英作文誤り訂正における複数の手法の利用に関する考察. 情報処理学会研究報告 自然言語処理研究会報告, Vol.2012-NL-208, pp.1-7.
- 水本智也, 小町守, 永田昌明 (NTT), 松本裕治. 文字ー単語アライメントを用いた日本語学習者の作文誤り訂正. 第 26 回人工知能学会全国大会論文集, pp.1-4, June 2012.
- 坂口慶祐, 水本智也, 小町守, 松本裕治. 英語スペリング訂正と品詞タグ付けの結合学習. 情報処理学会研究報告 自然言語処理研究会報告, Vol.2012-NL-206, pp.1-7, May 2012.
- 水本智也, 坂口慶祐, 小町守, 内海慶 (ヤフー), 河野洋志 (ヤフー), 前澤敏之 (ヤフー), 佐藤敏紀 (ヤフー). オークション検索クリックスルーログからの属性値抽出. 言語処理学会第 18 会年次大会 発表論文集, 1023-1026, March 2012.
- 藤野拓也, 水本智也, 小町守, 永田昌明 (NTT), 松本裕治. 日本語学習者の作文の誤り訂正に向けた単語分割. 言語処理学会第 18 回年次大会 発表論文集, pp.26-29, March 2012.
- 水本智也. 言語教育における自然言語処理の応用　ー 英語ストレス位置推定と誤り訂正 ー. 第 5 回言語学・自然言語処理合同勉強会. September 2011.
- 水本智也, 坂口慶祐. 学習者コーパスを用いた日本人英語誤り訂正. NLP 若手の会 第 6 回シンポジウム. September 2011.
- 水本智也, 小町守, 松本裕治. 大規模添削コーパスを用いた統計的機械翻訳手法による日本語誤り訂正. 言語処理学会第 17 回年次大会 発表論文集, pp.1007-1010, March 2011.
- 水本智也. Web から得られる大規模添削データを用いた機械翻訳手法による自動誤り訂正. 第 2 回入力メソッドワークショップ. December 2010.

賞

- Error Detection and Correction Workshop 2012 前置詞トラック 最優秀賞.
- Error Detection and Correction Workshop 2012 動詞トラック 優秀賞.
- Error Detection and Correction Workshop 2012 オープントラック 奨励賞.