Handling of Out-of-vocabulary Words in Japanese-English Machine Translation by Exploiting Parallel Corpus

Juan Luo and Yves Lepage
Graduate School of Information, Production and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Fukuoka 808-0135, Japan
juan.luo@suou.waseda.jp, yves.lepage@waseda.jp

Abstract

A large number of loanwords and orthographic variants in Japanese pose a challenge for machine translation. In this article, we present a hybrid model for handling out-of-vocabulary words in Japanese-to-English statistical machine translation output by exploiting parallel corpus. As the Japanese writing system makes use of four different script sets (kanji, hiragana, katakana, and romaji), we treat these scripts differently. A machine transliteration model is built to transliterate out-of-vocabulary Japanese katakana words into English words. A Japanese dependency structure analyzer is employed to tackle out-of-vocabulary kanji and hiragana words. The evaluation results demonstrate that it is an effective approach for addressing out-of-vocabulary word problems and decreasing the OOVs rate in the Japanese-to-English machine translation tasks.

Keywords

Out-of-vocabulary words, Machine translation, Parallel corpus, Machine transliteration, Text normalisation.

1. Introduction

Phrase-based statistical machine translation (PB-SMT) systems rely on parallel corpora for learning translation rules and phrases, which are stored in “phrase tables”. Words that cannot be found in phrase tables thus result in out-of-vocabulary words (OOVs) for a machine translation system. The large number of loanwords and orthographic variants in

Japanese makes the OOVs problem more severe than in other languages. As stated in (Oh et al., 2006), most of out-of-vocabulary words in translations from Japanese are made up of proper nouns and technical terms, which are phonetically transliterated from other languages. In addition, the highly irregular Japanese orthography as is analyzed in (Halpern, 2002) poses a challenge for machine translation tasks.

Japanese is written in four different sets of scripts: *kanji*, *hiragana*, *katakana*, and *romaji* (Halpern, 2002). *Kanji* is a logographic system consisting of characters borrowed from the Chinese characters. *Hiragana* is a syllabary system used mainly for functional elements. *Katakana* is also a syllabary system. Along with *hiragana*, both syllabaries are generally referred as *kana*. *Katakana* is used to write new words or loan words, i.e., words that are borrowed and transliterated from foreign languages. *Romaji* is just the Latin alphabet.

The problem of handling out-of-vocabulary words is not the major concern of machine translation literature. Traditional statistical machine translation systems either simply copy out-of-vocabulary words to the output, or bypass the problem by deleting these words in the translation output. Here we would like to stress that handling out-of-vocabulary words is important for the Japanese-to-English translation tasks.

We investigated the number of out-of-vocabulary words in the Japanese-to-English machine translation output. We built a standard PB-SMT system (see Section 5.3). The experiment was carried out by using a training set of 300,000 lines. The development set contains 1,000 lines, and 2,000 lines are used for test set. An analysis of the number of out-of-vocabulary words is presented in Table 1. In the output of a test set of 2,000 sentences, there are 237 out-of-vocabulary Japanese words. Among these OOV words, 96 out of 237 are *katakana* words, which is 40.51%. The number of OOV *kanji-hiragana* words is 141 (59.49%). It is observed from the output that 33 out of 141 OOV *kanji-hiragana* words (23.40%) are proper names. Therefore, further classification and treatment of *kanji-hiragana* words is needed.

	Data
Test sentences	2,000
Out-of-vocabulary words	237
OOV *katakana*	96
OOV *kanji-hiragana* (proper names)	33
OOV *kanji-hiragana* (others)	108

Table 1: Analysis of out-of-vocabulary words

In this article, we present a method to tackle out-of-vocabulary words to improve the performance of machine translation. This method makes use of two components. The first component deals with *katakana*. It relies on a machine transliteration model for *katakana* words that is based on the phrase-based machine translation framework. In addition, by making use of limited resources, i.e., the same parallel corpus used to build the machine translation system, a method of automatically acquiring bilingual word pairs for transliteration training data from this parallel corpus is used. With these enriched bilingual pairs, the transliteration model is further improved. The second component deals with *kanji-hiragana*. A Japanese dependency structure analyzer is used to build a *kanji-hiragana* system for handling orthographic variants.

The structure of the article is as follows. Section 2 reviews related works. Section 3 describes the first component. We present a back-transliteration model which is based on the SMT framework for handling *katakana* OOV words. Section 4 describes the second component and presents a method of tackling *kanji* and *hiragana* OOV words. Section 5 and 6 deal with the experiments and error analysis. Conclusion and future directions are drawn in Section 7.

2. Related Work

A number of works have been proposed to tackle the *katakana* out-of-vocabulary words by making use of machine transliteration. According to (Oh et al., 2006), machine transliteration can be classified into four models: grapheme-based transliteration model, phoneme-based transliteration model, hybrid transliteration model, and correspondence-based transliteration model.

A grapheme-based transliteration model tries to map directly from source graphemes to target graphemes (Li et al., 2004; Sherif and Kondrak, 2007; Garain et al., 2012; Lehal and Saini, 2012b). In the phoneme-based model, phonetic information or pronunciation is used, and thus an additional processing step of converting source grapheme to source phoneme is required. It tries to transform the source graphemes to target graphemes via phonemes as a pivot (Knight and Graehl, 1998; Gao et al., 2004; Ravi and Knight, 2009). A hybrid transliteration approach tries to use both the grapheme-based transliteration model and the phoneme-based model (Bilac and Tanaka, 2004; Lehal and Saini, 2012a). According to (Oh et al., 2006), the correspondence-based transliteration model (Oh and Choi, 2002) can also be considered as a hybrid approach. However, it differs from the others in that it takes into consideration the correspondence between a source grapheme and a source phoneme, while a general hybrid approach simply uses a combination of grapheme-based model and phoneme-based model through linear interpolation.

Machine transliteration, especially those methods that adopt statistical models, rely on training data to learn transliteration rules. Several studies on the automatic acquisition of transliteration pairs for different language pairs (e.g., English-Chinese, English-Japanese, and English-Korean) have been proposed in recent years.

Tsuji (2002) proposed a rule-based method of extracting *katakana* and English word pairs from bilingual corpora. A generative model is used to model transliteration rules, which are determined manually. As pointed out by Bilac and Tanaka (2005), there are two limitations of the method. One is the manually determined transliteration rules, which may pose the question of replication. The other is the efficiency problem of the generation of transliteration candidates. Brill et al. (2001) exploited non-aligned monolingual web search engine query logs to acquire *katakana*-English transliteration pairs. They firstly converted the *katakana* form to Latin script. A trainable noisy channel error model was then employed to map and harvest (*katakana*, English) pairs. The method, however, failed to deal with compounds, i.e., a single *katakana* word may match more than one English words. Lee and Chang (2003) proposed a statistical machine transliteration model to identify English-Chinese word pairs from parallel texts by exploiting phonetic similarities. Oh and Isahara (2006) presented a transliteration lexicon acquisition model to extract transliteration pairs from mining the web by relying on phonetic similarity and joint-validation.

While many techniques have been proposed to handle Japanese *katakana* words and translate these words into English, few works have focused on *kanji* and *hiragana*. As is shown in (Halpern, 2002), the Japanese orthography exhibits high variations, which contributes to a substantial number of out-of-vocabulary words in the machine translation output. A number of orthographic variation patterns have been analyzed by Halpern (2002): (1) okurigana variants, which are usually attached to a *kanji* stem; (2) cross-script orthographic variants, in which the same word can be written in a mixture of several scripts; (3) *kanji* variants, which can be written in different forms; (4) *kun* homophones, which means word pronounced the same but written differently.

In this article, we use a grapheme-based transliteration model to transform Japanese *katakana* out-of-vocabulary words to English, i.e., a model that maps directly from *katakana* characters to English characters without phonetic conversion. Furthermore, this model is used to acquire *katakana* and English transliteration word pairs from parallel corpus for enlarging the training data, which, in turn, improves the performance of the grapheme-based model. For handling *kanji* and *hiragana* out-of-vocabulary words, we propose to use a Japanese dependency structure analyzer and the source (i.e., Japanese) part of a parallel corpus to build a model for normalizing orthographic variants and translate them into English words.

3. Katakana OOV Model

Machine transliteration is the process of automatically converting terms in the source language into those terms that are phonetically equivalent in the target language. For example, the English word “chromatography” is transliterated in Japanese *katakana* word as “クロマトグラフィー” /ku ro ma to gu ra fi -/. The task of transliterating the Japanese words (e.g., クロマトグラフィー) back into English words (e.g., chromatography) is referred in (Knight and Graehl, 1998) as *back-transliteration*.

We view back-transliteration of unknown Japanese *katakana* words into English words as the task of performing character-level phrase-based statistical machine translation. It is based on the SMT framework as described in (Koehn et al., 2003). The task is defined as translating a Japanese *katakana* word $J_1^n = \{J_1,..., J_n\}$ to an English word $E_1^i = \{E_1,...E_i\}$, where each element of J_1^n and E_1^i is Japanese grapheme and English character. For a given Japanese *katakana J*, one tries to find out the most probable English word *E*. The process is formulated as

$$\arg\max_E P(E \mid J) = \arg\max_E P(J \mid E)P(E) \quad (1)$$

where *P(J|E)* is translation model and *P(E)* is the language model. Here the translation unit is considered to be graphemes or characters instead of words, and alignment is between graphemes and characters as is shown in Figure 1.

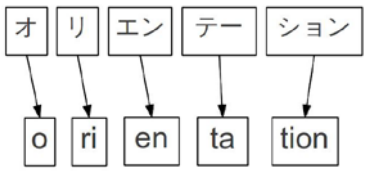

Figure 1: Character alignment

As the statistical model requires bilingual training data, a method of acquiring Japanese *katakana*-English word pairs from parallel corpus will be presented in the following section. The structure of the proposed method is summarized in Figure 2.

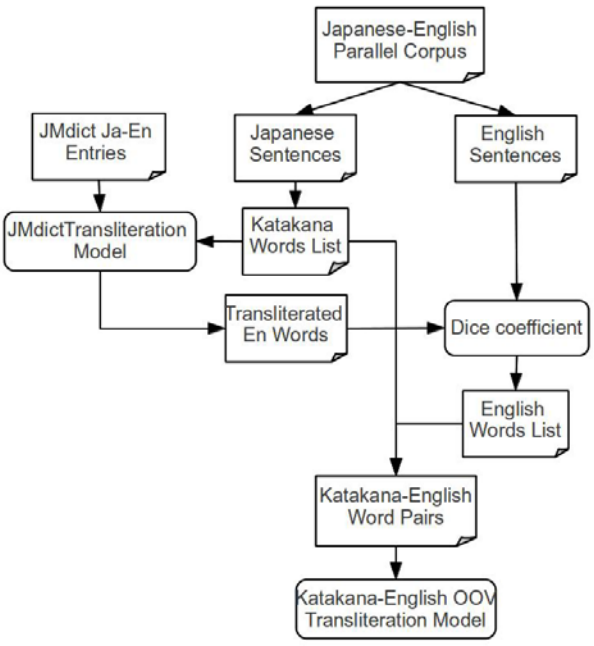

Figure 2: Illustration of katakana OOV model

3.1. Acquisition of Word Pairs

In this section, we will describe our method of obtaining *katakana*-English word pairs by making use of parallel corpus. The procedure consists of two stages. In the first stage, bilingual entries from a freely-available dictionary, JMdict (Japanese-Multilingual dictionary) (Breen, 2004), are firstly employed to construct a *seed* training data. By making use of this *seed* training set, a back-transliteration model that is based on the phrase-based SMT framework is then built. In the second stage, a list of *katakana* words is firstly extracted from the Japanese (source) part of the parallel corpus. These *katakana* words are then taken as the input of the back-transliteration model, which generate "transliterated" English words. After computing the Dice coefficient between the "transliterated" word and candidate words from the English (target) part of the parallel corpus, a list of pairs of *katakana*-English words is finally generated.

To measure the similarities between the transliterated word w_x and target candidate word w_y , the Dice coefficient (Dice, 1945) is used. It is defined as

$$Dice(w_x, w_y) = \frac{2 \times n(w_x, w_y)}{n(w_x) + n(w_y)} \quad (2)$$

where $n(w_x)$and $n(w_y)$ are the number of bigram occurrences in word w_x and w_y respectively, and $n(w_x , w_y)$ represents the number of bigram occurrences found in both words.

3.1.1. One-to-many Correspondence

There is the case where a single *katakana* word may match a sequence of English words. This is a problem identified in previous research (Brill et al., 2001). Examples are shown in Table 2. In order to take into consideration one-to-many matches and extract those word pairs from parallel corpus, we preprocessed the English part of the corpus. Given a *katakana* word, for its counterpart, the English sentence, we segment it into n-grams, where $n \leq 3$. The Dice coefficient is then calculated between the "transliterated" word of this *katakana* and English n-grams (i.e., unigrams, bigrams, and trigrams) to measure the similarities. This method allows to harvest not only one-to-one but also one-to-many (*katakana*, English) word pairs from parallel corpus.

Katakana	English
トナーパターン	toner pattern
フラッシュメモリ	flash memory
アイスクリーム	ice cream
グラフィックユーザインタフェース	graphic user interface
デジタルシグナルプロセッサ	digital signal processor
プロダクトライフサイクル	product life cycle

Table 2: One-to-many correspondence

4. Kanji-hiragana OOV Model

Japanese is written in four scripts (*kanji*, *hiragana*, *katakana*, and *romaji*). The use of this set of scripts in a mixture causes the high orthographical variation. As analyzed in (Halpern, 2002), there are a number of patterns: okurigana variants, cross-script orthographic variants, kana variants, orthographic ambiguity for *kun* homophones written in hiragana, and so on. Table 3 shows an example of okurigana variants and *kun* homophones. These Japanese orthographic variants pose a special challenge for machine translation tasks.

Patterns	English	Reading	Variants	Phonetics
Okurigana variants	‘moving’	/hikkoshi/	引越し 引っ越し 引越	ヒッコシ
	‘effort’	/torikumi/	取り組み 取組み 取組	トリクミ
Kun homophones	‘bridge’ ‘chopsticks’	/hashi/	橋 箸	ハシ ハシ
	‘account’ ‘course’	/kouza/	口座 講座	コウザ コウザ

Table 3: Orthographic variants

In this section, we will present our approach for tackling and normalizing out-of-vocabulary *kanji* and *hiragana* words. These words are classified into two categories: proper names and other *kanji-hiragana* OOVs.

To handle proper names, we firstly obtain their phonetic forms by using a Japanese dependency structure analyzer. Then, we employ the Hepburn romanization charts (i.e., a mapping table between characters and the Latin alphabet) to transform these named entities into English words. Let us illustrate the approach with an example. Assume there is a OOV word “藤木”, which is a personal name. The dependency structure analyzer is applied to generate its phonetic form “フジキ”. By referring to the Hepburn romanization charts, we then simply transform its phonetic form into English words “Fujiki”.

The architecture of the approach to handle *kanji-hiragana* OOVs except for proper names is summarized in Figure 3. The method comprises two processes: (a) building a model; (b) normalizing and translating *kanji-hiragana* OOVs. In the first process, we use the Japanese part of the parallel corpus (the same Japanese-English parallel corpus used for training in the standard phrase-based SMT) as the input to the Japanese dependency structure analyzer CaboCha (Kudo and Matsumoto, 2002). A phonetic-to-standard Japanese parallel corpus (Figure 4) is then obtained to train a monolingual Japanese model which is also built upon a phrase-based statistical machine translation framework. In the second process, the dependency structure analyzer is applied to generate corresponding phonetics from a list of *kanji-hiragana* out-of-vocabulary words. These OOVs in the phonetic forms are then input to the monolingual model to produce a list of normalized *kanji-hiragana* words. Finally, the normalized OOV words will be translated into English.

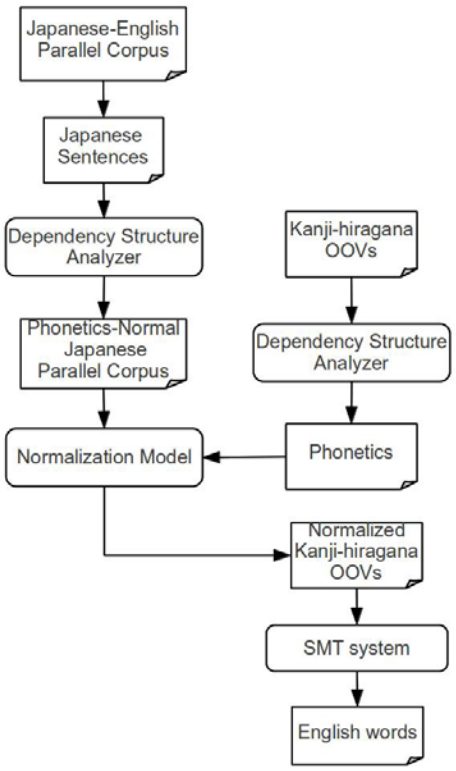

Figure 3: Illustration of kanji-hiragana OOV model

フカ オウトウ セイギョ ヲ	負荷 応答 制御 を
アイドル カイテン スウ セイギョ ヲ	アイドル 回転 数 制御 を
ティック ワリコミ ショリ ガ	ティック 割り込み 処理 が
ホウデン デンリュウ ヲ	放電 電流 を
カイシ サ レル。	開始 さ れる 。
モール ド ブ デ アル。	モール ド 部 で ある 。
シュツリョク ベースアドレス ヲ	出力 ベースアドレス を
ドウヨウ ナイブ シンゴウ モ	同様 内部 信号 も
ゲンテイ サ レル	限定 さ れる
キリカエル	切り換える
クミアワセ ニヨル	組み合わせ による

Figure 4: Sample of phonetic-to-standard Japanese parallel corpus

5. Experiments

In this section, we present the results of three experiments. In the first experiment, we evaluate the performance of the back-transliteration model. The data sets used in the back-transliteration system comprise one-to-one or one-to-many *Katakana*-English word pairs, which are segmented at the character level. In the second experiment, the performance of the model for normalizing *kanji-hiragana* is assessed. In the third setting, the performance of handling both *katakana* and *kanji-hiragana* out-of-vocabulary words in a machine translation output will be evaluated. The first two experiments are thus intrinsic evaluation experiments, while the last one, which assesses our proposed method by measuring its contribution to a different task, is an extrinsic evaluation experiment.

5.1. Katakana Transliteration Test

To train a back-transliteration model which is built upon a phrase-based statistical machine translation framework, we used the state-of-the-art machine translation toolkit: Moses decoder (Koehn et al., 2007), alignment tool GIZA++ (Och and Ney, 2003), MERT (Minimum Error Rate Training) (Och, 2003) to tune the parameters, and the SRI Language Modeling toolkit (Stolcke, 2002) to build character-level target language model.

The data set for training (499,871 entries) we used in the experiment contains the JMdict entries and word pairs extracted from parallel corpus. The JMdict consists of 166,794 Japanese-English entries. 19,132 *katakana*-English entries are extracted from the dictionary. We also extracted 480,739 *katakana*-English word pairs from NTCIR Japanese-English parallel corpus. The development set is made of 500 word pairs, and 500 entries are used for test set. To train back-transliteration models, transliteration pairs between Japanese and English like the ones provided by the NEWS workshop or distributed by the Linguistic Data Consortium (LDC) could be used[1].

The experimental results are shown in Table 4. For evaluation metric, we used BLEU at the character level (Papineni et al., 2002; Denoual and Lepage, 2005; Li et al., 2011). Word accuracy and character accuracy (Karimi et al., 2011) are also used to assess the performance of the system. Word accuracy (WA) is calculated as:

$$WA = \frac{\text{number of correct transliterations}}{\text{total number of test words}} \tag{3}$$

Character accuracy (CA) is based on the Levenshtein edit distance (Levenshtein, 1966) and it is defined as:

[1]http://translit.i2r.a-star.edu.sg/news2012/. As such data are not freely available or require to subscribe, we did not use such data in our experiments. Thanks to the reviewers for pointing such data.

$$CA = \frac{len(T) - ED(T, L(T_i))}{len(T)} \quad (4)$$

where *len(T)* is the length of reference word *T*. $L(T_i)$ is the best suggested transliteration, and *ED* is the Levenshtein edit distance (insertion, deletion, and substitution) between two words. The character accuracy takes an average of all the test entries.

System	BLEU	WA	CA
Katakana transli.	80.56	50.60%	86.33%

Table 4: Evaluation results of transliteration test

An analysis of number of character errors in entry strings is shown in Table 5. 253 out of 500 entries (50.60%) match exactly the same as the reference words. The number of strings that contain one or two character errors are 86 (17.20%) and 56 (11.20%), respectively. In total, strings with less than two character errors represent 79.00% of all the test entries. There are 50 (10.00%) and 55 (11.00%) entries containing three or more than three character errors.

Character errors	Entries	Percentage
0 character error	253	50.60%
1 character error	86	17.20%
2 character error	56	11.20%
3 character error	50	10.00%
Others	55	11.00%

Table 5: Analysis of number of character errors

Examples of *katakana*-English transliteration output are given in Table 6. For some *katakana* words, they are transliterated correctly as references. For other *katakana* words, it shows that the output of transliteration contains spelling errors. For example, the grapheme “アン” can be transliterated into “an”, “en”, or “un”. For the *katakana* word “アンハッピー” (unhappy), it is erroneously transliterated into “anhappy”.

5.2. Kanji-hiragana Normalization Test

In the second experiment, we assess the performance of *kanji-hiragana* normalization model as described in Section 4. As the monolingual Japanese normalization model is also built upon the statistical machine translation framework, we use the same toolkit as in

	Katakana	Reference	Output
0	インベンション	invention	invention
0	インプット	input	input
0	アンカー	anchor	anchor
1	アンカーマン	anchorman	ancherman
1	アンハッピー	unhappy	anhappy
1	アントレ	entree	entre
2	インテルクチュアル	intellectual	intelctual
2	インビジブル	invisible	inbsible
2	インテリア	interior	interia
n	インターフェアランス	interference	interfealance
n	アンフェア	unfair	anfare
n	アンタッチャブル	untouchable	antatchable

Table 6: Transliteration output examples sorted by number of character errors

Section 5.1. For the training set, we apply the Japanese dependency structure analyzer CaboCha on the Japanese part of the parallel corpus (300,000 lines) and obtain a phonetic-to-standard Japanese parallel corpus (see Figure 4). The development set and test set consist of 1,000 lines and 5,000 words, respectively. Since this experiment is not a task of measuring the accuracy of the output of the model (i.e., it is a test of how the monolingual model can normalize the Japanese *kanji-hiragana* words), we did not use any evaluation metrics, such as BLEU, WA, and CA.

Table 7 shows an analysis of number of character differences between *kanji-hiragana* words and their normalized forms. The number of entries matches exactly the same as the original Japanese words is 3908, which represents 78.16% of all test entries. There are 21.84% of the entries which are normalized to different forms. Examples of number of character differences are shown in Table 8. The normalized output forms can generally be categorized into three types: kun homophones, okurigana variants, and others. Kun homophones would cause orthographic ambiguity. Words in the category okurigana variants are normalized into different forms but they have the same meaning. It shows that the monolingual normalization model is useful for solving out-of-vocabulary okurigana variants and helps reducing the out-of-vocabulary words rate. In an SMT system, this will reduce the number of types. There are other words that are not normalized for which the phonetic representation is output directly.

No. of character diff.	Entries	Percentage
0	3,908	78.16%
1	424	8.48%
2	509	10.18%
3	44	0.88%
more than 3	115	2.30%

Table 7: Analysis of number of character differences

	Japanese	Phonetics	Norm. output
0	駐車 (parking)	チュウシャ	駐車 (parking)
0	飲み物 (beverage)	ノミモノ	飲み物 (beverage)
0	電極 (electrode)	デンキョク	電極 (electrode)
kun homophones			
1	視点 (perspective)	シテン	支点 (fulcrum)
1	通貨 (currency)	ツウカ	通過 (pass)
1	講座 (course)	コウザ	口座 (account)
2	注視 (gaze)	チュウシ	中止 (stop)
2	意思 (intention)	イシ	医師 (doctor)
2	近郊 (suburbs)	キンコウ	均衡 (balance)
n	当たり (per)	アタリ	辺 (side)
okurigana variants			
1	読みとり (read)	ヨミトリ	読み取り
1	繰返し (repeat)	クリカエシ	繰り返し
1	呼出し (call)	ヨビダシ	呼び出し
2	纏め (collect)	マトメ	まとめ
2	釣合 (balance)	ツリアイ	釣り合い
2	振替 (transfer)	フリカエ	振り替え
n	うま味 (umami)	ウマミ	旨み
others			
n	切替 (switch)	キリカエ	切り換え
n	雪崩 (avalanche)	ナダレ	ナダレ
n	藤木 (personal name)	フジキ	フジキ

Table 8: Examples of character differences can be seen by comparing the Japanese column with the Normalized output column

5.3. Out-of-vocabulary Words Test

In the third experiment, we evaluate the performance of handling out-of-vocabulary words for machine translation by making use of *katakana* OOV model and *kanji-hiragana* OOV model. The system architecture is summarized in Figure 5. From the output of a machine translation system, out-of-vocabulary words are firstly extracted. OOV *katakana* words are then transliterated into English by using the back-transliteration model and OOV *kanji-hiragana* words are normalized and translated into English words by using the normalization model. A standard phrase-based statistical machine translation system is built by making use of the same toolkit as described in Section 5.1. KyTea (Neubig et al., 2011) is used to perform segmentation on *katakana* OOV words.

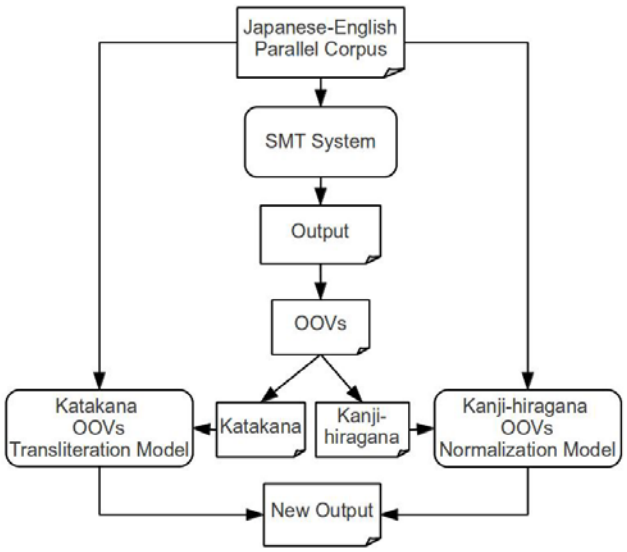

Figure 5: Illustration of system architecture

For data sets in the baseline SMT system, we use a sample of NTCIR Japanese-English parallel corpus. The training set is made of 300,000 lines. The development set contains 1,000 lines, and 2,000 lines are used for test set.

As for the evaluation, while the quality of a machine translation system is usually measured in BLEU scores, it may not be fair to examine the results in BLEU scores for measuring the improvement and contribution of out-of-vocabulary *katakana* transliteration and *kanji-hiragana* normalization to a machine translation system. Here we provide the BLEU scores as a reference. Table 9 shows the evaluation results of OOV words test. By

comparing with the baseline system, it shows that there is a slight gain in BLEU for transliterating out-of-vocabulary *katakana* words and normalizing and translating *kanji-hiragana* words. We also extracted sentences that contain out-of-vocabulary words (166 lines) from the test set. In comparison with the baseline, sentences with translated out-of-vocabulary words give better scores.

System	BLEU
Japanese-English MT baseline	25.25
MT with katakana OOV model	25.28
MT with kanji-hiragana OOV model	25.27
MT with both models	25.30
Sentences with OOV (MT baseline)	14.67
Sentences with OOV (both models)	15.14

Table 9: Evaluation results of OOV words test

6. Error Analysis

To summarize, all our experiments, the main points observed from a scrutinous analysis of the results of *katakana* OOV model and *kanji-hiragana* OOV model and countermeasures against them are as follows:

Katakana OOV model: by examining the output, we analyze the number of character errors (see Table 10). From the table, we can see that almost 62% of edit operations are insertions. Substitution accounts for 35% of edit operations. Among 61 erroneous transliterated *katakana* words, 26 may have been caused by a wrong segmentation. These compound *katakana* words are not segmented appropriately, which result in erroneous English transliteration. Further improvement on back-transliteration model would be expected when the accuracy of segmentation of *katakana* words is improved.

- the word: インストルメンタルパネル (instrumental panel)
  wrong segmentation: インストル | メンタル | パネル
  correct segmentation: インストルメンタル | パネル
  transliteration result: instru mental panel

- the word: ウエハホルダ (wafer holder)
  wrong segmentation: ウエハホルダ

correct segmentation: ウエハ | ホルダ
transliteration result: waferholder

	Edit	Number	Percentage
Insertion	1	31	38.27%
	2	8	9.88%
	> 2	11	13.58%
Deletion	1	3	3.70%
	2	0	0.00%
	> 2	0	0.00%
Substitution	1	10	12.35%
	2	12	14.81%
	> 2	6	7.41%

Table 10: Analysis of character errors

Kanji-hiragana OOV model: handling *kanji-hiragana* words is very difficult due to the orthographic variants and the complexity of the Japanese writing system. As a positive result, the model can handle named entities. For instance, the personal name “吉崎” is transformed phonetically into “ヨシザキ” and translated correctly into “Yoshizaki”. The model is also useful for handling okurigana variants. For example, the word “閉込め” is normalized into “閉じ込め” and translated correctly into “confinement”. However, as a negative result, some of the normalized *kanji-hiragana* words cannot be translated correctly into English words. Here, we analyze and categorize into three different kinds of errors:

1) Among 95 erroneous normalized *kanji-hiragana* words, 31.58% (30) of words are normalized into their original form, i.e., they match exactly the same as the original Japanese words and cannot be translated into English.

- kanji: 馬術 (equestrianism)
  phonetics: バジュツ
  normalize: 馬術
  translation: 馬術

2) 38.95% (37) of words are output directly in their phonetic representation form.

- kanji: 過渡期 (transition period)
  phonetics: カトキ
  normalize: カトキ
  translation: カトキ

3) There are 29.47% (28) of *kanji-hiragana* words are normalized to different forms, which result in erroneous translation to English words. These words are normalized and transformed into different written forms as they are pronounced the same (homophones), which leads to wrong translation.

- kun homophones: 変事 (accident)
  phonetics: ヘンジ
  normalize: 返事 (reply)
  translation: reply

- kun homophones: 高配 (trouble)
  phonetics: コウハイ
  normalize: 荒廃 (ruins)
  translation: ruins

7. Conclusion and Future Work

We have described a method of handling both *katakana* and *kanji-hiragana* out-of-vocabulary words by exploiting a parallel corpus. A grapheme-based back-transliteration model is built upon the phrase-based statistical machine translation framework for transliterating *katakana* words into English words. This model is also used to enrich the training set by extracting Japanese *katakana* and English word pairs from parallel corpus. A normalization model is built to tackle and translate *kanji-hiragana* words. While there are limitations of the model, it can be an aid to normalize and translate okurigana variants and proper names.

The experimental results reveal that segmentation of Japanese katakana words should be improved, which will be our future work.

8. References

Bilac, S. and Tanaka, H., 2004, A hybrid back-transliteration system for Japanese, In *Proceedings of the 20th international conference on Computational Linguistics (COLING 2004)*, pp. 597–603.

Bilac, S. and Tanaka, H., 2005, Extracting transliteration pairs from comparable corpora, In *Proceedings of the Annual Meeting of the Natural Language Processing Society*, Japan.

Breen, J., 2004, Jmdict: a Japanese – Multilingual dictionary, In *Proceedings of the Coling 2004Workshop on Multilingual Linguistic Resources*, pp. 71–78, Geneva.

Brill, E., Kacmarcik, G., and Brockett, C., 2001, Automatically harvesting katakana-English term pairs from search engine query logs, In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPRS 2001)*, pp. 393–399, Tokyo, Japan.

Denoual, E. and Lepage Y., 2005, BLEU in characters: towards automaticMT evaluation in languages without word delimiters, In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP 2005)*, pp. 79–84, Jeju Island, Republic of Korea, October.

Dice, L. R., 1945, Measures of the amount of ecologic association between species. *Journal of Ecology*, 26(3):297–302.

Gao, W., Wong, K. F., and Lam, W., 2004, Phoneme-based transliteration of foreign names for OOV problem, In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP 2004)*, pp. 110–119, Berlin, Heidelberg.

Garain, U., Das, A., Doermann, D., and Oard, D., 2012, Leveraging statistical transliteration for dictionary-based English-Bengali CLIR of OCR'd text, In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pp. 339–348, Mumbai, India, December.

Halpern, J., 2002, Lexicon-based orthographic disambiguation in CJK intelligent information retrieval, In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization*, pp. 1–7.

Karimi, S., Scholer, F., and Turpin, A., 2011, Machine transliteration survey. *ACM Computing Surveys*, 43(3):17:1–17:46, April.

Knight, K. and Graehl, J., 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612, December.

Koehn, P., Och, F. J., and Marcu, D., 2003, Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pp. 48–54, Edmonton.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E., 2007, Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pp. 177–180, Prague, Czech Republic.

Kudo, T. and Matsumoto, Y., 2002, Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pp. 63–69, Taipei, Taiwan.

Lee, C. J. and Chang, J. S., 2003, Acquisition of English-Chinese transliterated word pairs from parallel-aligned texts using a statistical machine transliteration model. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pp. 96–103.

Lehal, G. S. and Saini, T. S., 2012a, Conversion between scripts of Punjabi: Beyond simple transliteration. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pp. 633–642, Mumbai, India, December.

Lehal, G. S. and Saini, T. S., 2012b, Development of a complete Urdu-Hindi transliteration system. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pp. 643–652, Mumbai, India, December.

Levenshtein, V. I., 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-doklady*, 10(8):707–710.

Li, H. Z., Zhang, M., and Su, J., 2004, A joint source-channel model for machine transliteration. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL 2004)*, pp. 159–166.

Li, M. X., Zong, C. Q., and Ng, H. T., 2011, Automatic evaluation of Chinese translation output: word-level or character-level? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pp. 159–164, Portland, Oregon, USA.

Neubig, G., Nakata, Y., and Mori, S., 2011, Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pp. 529–533, Portland, Oregon, USA.

Och, F. J. and Ney, H., 2003, A systematic comparison of various statistical alignment models, *Computational Linguistics*, 29(1):19–51.

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

Oh, J. H. and Choi, K. S., 2002, An English-Korean transliteration model using pronunciation and contextual rules. In *Proceedings of the 19th International Conference on Computational linguistics (COLING 2002)*, pp. 1–7.

Oh, J. H. and Isahara, H., 2006, Mining the web for transliteration lexicons: Joint-validation approach, In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 254–261, Washington, DC, USA.

Oh, J. H., Choi, K. S., and Isahara, H., 2006, A comparison of different machine transliteration models, *Journal of Artificial Intelligence Research*, 27(1):119–151, October.

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

Ravi, S. and Knight, K., 2009, Learning phoneme mappings for transliteration without parallel data, In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2009)*, pp. 37–45.

Sherif, T. and Kondrak, G., 2007, Substring based transliteration. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pp. 944–951, Prague, Czech Republic, June.

Stolcke, A., 2002, SRILM-an extensible language modeling toolkit, In *Proceedings of the Seventh International Conference on Spoken Language Processing (ICSLP 2002)*, volume 2, pp. 901–904, Denver, Colorado.

Tsuji, K., 2002, Automatic extraction of translational Japanese-katakana and English word pairs from bilingual corpora, *International Journal of Computer Processing of Oriental Languages*, 15(3):261–279.