Hypothesis Selection in Machine Transliteration: A Web Mining Approach

Jong-Hoon Oh and Hitoshi Isahara
Computational Linguistics Group
National Institute of Information and Communications Technology (NICT)
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{rovellia, isahara}@nict.go.jp

Abstract
We propose a new method of selecting hypotheses for machine transliteration. We generate a set of Chinese, Japanese, and Korean transliteration hypotheses for a given English word. We then use the set of transliteration hypotheses as a guide to finding relevant Web pages and mining contextual information for the transliteration hypotheses from the Web page. Finally, we use the mined information for machine-learning algorithms including support vector machines and maximum entropy model designed to select the correct transliteration hypothesis. In our experiments, our proposed method based on Web mining consistently outperformed systems based on simple Web counts used in previous work, regardless of the language.

1 Introduction
Machine transliteration has been a great challenge for cross-lingual information retrieval and machine translation systems. Many researchers have developed machine transliteration systems that accept a source language term as input and then output its transliteration in a target language (Al-Onaizan and Knight, 2002; Goto et al., 2003; Grefenstette et al., 2004; Kang and Kim, 2000; Li et al., 2004; Meng et al., 2001; Oh and Choi, 2002; Oh et al., 2006; Qu and Grefenstette, 2004). More precisely, they used simple Web counts, estimated as the number of hits (Web pages) retrieved by a Web search engine.

However, there are several limitations imposed on the ability of Web counts to select a correct transliteration hypothesis. First, the assumption that hit counts approximate the Web frequency of a given query usually introduces noise (Lapata and Keller, 2005). Moreover, some Web search engines disregard punctuation and capitalization when matching search terms (Lapata and Keller, 2005). This can cause errors if such Web counts are relied on to select transliteration hypotheses. Second, it is not easy to consider the contexts of transliteration hypotheses with Web counts because Web counts are estimated based on the number of retrieved Web pages. However, as our preliminary work showed (Oh et al., 2006), transliteration or translation pairs often appear as parenthetical expressions or tend to be in close proximity in texts; thus context can play an important role in selecting transliteration hypotheses. For example, there are several Chinese, Japanese, and Korean (CJK) transliterations and their counterparts in a parenthetical expression, as follows.

1) 阿德里安娜1克拉克森2 (Adrienne1 Clarkson2)
2) グルコース1 オキシダーゼ2 (glucose1 oxidase2)
3) ダフェネ1 オキシダーゼ2 (diphenol1 oxidase2)

Note that the subscripted numbers in all examples represent the correspondence between the English word and its CJK counterpart. These parenthetical expressions are very useful in selecting translit-
eration hypotheses because it is apparent that they are translation pairs or transliteration pairs. However, we cannot fully use such information with Web counts.

To address these problems, we propose a new method of selecting transliteration hypotheses. We were interested in how to mine information relevant to the selection of hypotheses and how to select correct transliteration hypotheses using the mined information. To do this, we generated a set of CJK transliteration hypotheses for a given English word. We then used the set of transliteration hypotheses as a guide to finding relevant Web page and mining contextual information for the transliteration hypotheses from the Web page. Finally, we used the mined information for machine-learning algorithms including support vector machines (SVMs) and maximum entropy model designed to select the correct transliteration hypothesis.

This paper is organized as follows. Section 2 describes previous work based on simple Web counts. Section 3 describes a way of generating transliteration hypotheses. Sections 4 and 5 introduce our methods of Web mining and selecting transliteration hypotheses. Sections 6 and 7 deal with our experiments and the discussion. Conclusions are drawn and future work is discussed in Section 8.

2 Related work

Web counts have been used for selecting transliteration hypotheses in several previous work (Al-Onaizan and Knight, 2002; Grefenstette et al., 2004; Oh et al., 2006; Qu and Grefenstette, 2004). Because the Web counts are estimated as the number of hits by a Web search engine, they greatly depend on queries sent to a search engine. Previous work has used three types of queries—monolingual queries (MQs) (Al-Onaizan and Knight, 2002; Grefenstette et al., 2004; Oh et al., 2006), bilingual simple queries (BSQs) (Oh et al., 2006; Qu and Grefenstette, 2004), and bilingual bigram queries (BBQs) (Oh et al., 2006). If we let $S$ be a source language term and $\mathcal{H} = \{h_1, \ldots, h_r\}$ be a set of machine-generated transliteration hypotheses of $S$, the three types of queries can be defined as

**MQ:** $h_i$ (e.g., 克林頓, Clinton, and Clinton 푘러린턴).

**BSQ:** $s$ and $h_i$ without quotations (e.g., Clinton 克林頓, Clinton クリントン, and Clinton 클러린턴).

**BBQ:** Quoted bigrams composed of $S$ and $h_i$ (e.g., “Clinton 克林頓”, “Clinton クリントン”, and “Clinton 클러린턴”).

MQ is not able to determine whether $h_i$ is a counter-part of $S$, but whether $h_i$ is a frequently used target term in target-language texts. BSQ retrieves Web pages if $S$ and $h_i$ are present in the same document but it does not take the distance between $S$ and $h_i$ into consideration. BBQ retrieves Web pages where “$S h_i$” or “$h_i S$” are present as a bigram. The relative order of Web counts over $\mathcal{H}$ makes it possible to select transliteration hypotheses in the previous work.

3 Generating Transliteration Hypotheses

Let $S$ be an English word, $P$ be a pronunciation of $S$, and $T$ be a target language transliteration corresponding to $S$. We implement English-to-CJK transliteration systems based on three different transliteration models—a grapheme-based model ($S \rightarrow T$), a phoneme-based model ($S \rightarrow P$ and $P \rightarrow T$), and a correspondence-based model ($S \rightarrow P$ and $(S, P) \rightarrow T$) as described in our preliminary work (Oh et al., 2006). $P$ and $T$ are segmented into a series of sub-strings, each of which corresponds to a source grapheme. We can thus write $S = s_1, \ldots, s_n = s_n^1, P = p_1, \ldots, p_n = p_n^1$, and $T = t_1, \ldots, t_n = t_n^1$, where $s_i$, $p_i$, and $t_i$ represent the $i$th English grapheme, English phonemes corresponding to $s_i$, and target language graphemes corresponding to $s_i$, respectively. Given $S$, our transliteration systems generate a sequence of $t_i$ corresponding to either $s_i$ (in Eq. (1)) or $p_i$ (in Eq. (2)) or both of them (in Eq. (3)).

$$Pr_G(T|S) = Pr(t_n^1|s_n^1)$$  \hspace{1cm} (1)
$$Pr_P(T|S) = Pr(p_n^1|s_n^1) \times Pr(t_n^1|p_n^1)$$  \hspace{1cm} (2)
$$Pr_C(T|S) = Pr(p_n^1|s_n^1) \times Pr(t_n^1|s_n^1, p_n^1)$$  \hspace{1cm} (3)

The maximum entropy model was used to estimate probabilities in Eqs. (1)–(3) (Oh et al., 2006). We produced the $n$-best transliteration hypotheses using a stack decoder (Schwartz and Chow, 1990). We
then created a set of transliteration hypotheses comprising the $n$-best transliteration hypotheses.

## 4 Web Mining

Let $S$ be an English word and $\mathcal{H} = \{h_1, \ldots, h_r\}$ be its machine-generated set of transliteration hypotheses. We use $S$ and $\mathcal{H}$ to generate queries sent to a search engine\(^1\) to retrieve the top-100 snippets. A correct transliteration and its counterpart tend to be in close proximity on CJK Web pages. Our goal in Web mining was to find such Web pages and mine information that would help to select transliteration hypotheses from these pages.

To find these Web pages, we used three kinds of queries, $Q_1 = (S$ and $h_1)$, $Q_2 = S$, and $Q_3 = h_i$, where $Q_1$ is the same as BSQ’s query and $Q_3$ is the same as MQ’s. The three queries usually result in different sets of Web pages. We categorize the retrieved Web pages by $Q_1$, $Q_2$, and $Q_3$ into $W_1$, $W_2$, and $W_3$. We extract three kinds of features from $W_l$ as follows, where $l = 1, 2, 3$.

- $Freq(h_i, W_l)$: the number of occurrences of $h_i$ in $W_l$
- $DFreq(h_i, W_l)$: Co-occurrence of $S$ and $h_i$ with distance $d_k \in D$ in the same snippet of $W_l$.
- $PFreq(h_i, W_l)$: Co-occurrence of $S$ and $h_i$ as parenthetical expressions with distance $d_k \in D$ in the same snippet of $W_l$. Parenthetical expressions are detected when either $S$ or $h_i$ is in parentheses.

We define $D = \{d_1, d_2, d_3\}$ with three ranges of distances between $S$ and $h_i$, where $d_1 (d < 5)$, $d_2 (5 \leq d < 10)$, and $d_3 (10 \leq d \leq 15)$. We counted distance $d$ with the total number of characters (or words)\(^2\) between $S$ and $h_i$. Here, we can take the contexts of transliteration hypotheses into account using $DFreq$ and $PFreq$; while $Freq$ is counted regardless of the contexts of the transliteration hypotheses.

Figure 1 shows examples of how to calculate $Freq$, $DFreq_k$, and $PFreq_k$, where $S = Clinton$.

\(^1\)We used Google (http://www.google.com)

\(^2\)Depending on whether the languages had spacing units, words (for English and Korean) or characters (for Chinese and Japanese) were chosen to calculate $d$.

Figure 1: Web corpora collected by Clinton and 克林頓

| Snippet1 | 克林頓1 | 克林頓2 | 克林頓3 |
|----------|--------|--------|--------|
| Clinton1 | 1      | 41     | 68     |
| Clinton2 | 72     | 29     | 2      |

| Snippet2 | 克林頓4 | 克林頓5 | 克林頓6 |
|----------|--------|--------|--------|
| Clinton3 | 0      | 36     | 81     |
| Clinton4 | 40     | 0      | 37     |
| Clinton5 | 85     | 41     | 0      |

| Snippet2 | 克1  | 克2  | 克3  |
|----------|-----|-----|-----|
| Clinton3 | 6   | 9   | 85  |
| Clinton4 | 32  | 29  | 42  |
| Clinton5 | 77  | 74  | 1   |

Table 1: Distance between Clinton and Chinese transliteration hypotheses in Fig. 1

$h_i$: 克林頓 in $W_1$ collected by $Q_1 = (Clinton$ 克林頓). The subscripted numbers of Clinton and 克林頓 were used to indicate how many times they occurred in $W_1$. In Fig. 1, 克林頓 occurs six times thus $Freq(h_i, W_1) = 6$. Table 1 lists the distance between Clinton and 克林頓 within each snippet of $W_1$. We can obtain $DFreq_1(h_i, W_1)$ = 5. $PFreq_1(h_i, W_1)$ is calculated by detecting parenthetical expressions between $S$ and $h_i$ when $DFreq_1(h_i, W_1)$ is counted. Because all $S$ in $W_1$ (Clinton1 to Clinton5) are in parentheses, $PFreq_1(h_i, W_1)$ is the same as $DFreq_1(h_i, W_1)$.

We ignore $Freq$, $DFreq_k$, and $PFreq_k$ when $h_i$ is a substring of other transliteration hypotheses because $h_i$ usually has a higher $Freq$, $DFreq_k$, and $PFreq_k$ than $h_j$ if $h_i$ is a substring of $h_j$. Let a
set of transliteration hypotheses for $S = Clinton$ be $H = \{h_1 = \text{Clinton}, h_2 = \text{Cl}\}$. Here, $h_2$ is a substring of $h_1$. In Fig. 1, $h_2$ appears six times as a substring of $h_1$ and three times independently in Snippet2. Moreover, independently used $h_2 (\text{Clinton}_1, \text{Clinton}_2, \text{Clinton}_3$) and $S (\text{Clinton}_3$ and $\text{Clinton}_5$) are sufficiently close to count $DFreq_k$ and $PFreq_k$. Therefore, the $Freq$, $DFreq_k$, and $PFreq_k$ of $h_1$ will be lower than those of $h_2$ if we do not take the substring relation between $h_1$ and $h_2$ into account. Considering the substring relation, we obtain $Freq(h_2, W_1) = 3$, $DFreq(h_2, W_1) = 1$, $DFreq(h_2, W_1) = 2$, $PFreq(h_2, W_1) = 1$, and $PFreq(h_2, W_1) = 2$.

5 Hypothesis Selection

We select transliteration hypotheses by ranking them. A set of transliteration hypotheses, $H = \{h_1, h_2, \ldots, h_r\}$, is ranked to enable a correct hypothesis to be identified. We devise a rank function, $g(h_i)$ in Eq. (4), that ranks a correct transliteration hypothesis higher and the others lower.

$$g(h_i) : H \rightarrow \{R : R \text{ is ordering of } h_i \in H\}$$ (4)

Let $x_i \in X$ be a feature vector of $h_i \in H$, $y_i = \{+1, -1\}$ be the training label for $x_i$, and $TD = \{td_1 = <x_1, y_1>, \ldots, td_r = <x_2, y_2>\}$ be the training data for $g(h_i)$. We prepare the training data for $g(h_i)$ as follows.

1. Given each English word $S$ in the training-set, generate transliteration hypotheses $H$.

2. Given $h_i \in H$, assign $y_i$ by looking for $S$ and $h_i$ in the training-set — $y_i = +1$ if $h_i$ is a correct transliteration hypothesis corresponding to $S$, otherwise $y_i = -1$.

3. For each pair $(S, h_i)$, generate its feature vector $x_i$.

4. Construct a training data set, $TD$:
   - $TD = TD^+ \cup TD^-$
   - $TD^+ \ni td_i$ where $y_i = +1$
   - $TD^- \ni td_j$ where $y_j = -1$

   We used two machine-learning algorithms, support vector machines (SVMs)\(^3\) and maximum entropy model\(^4\) for our implementation of $g(h_i)$. The SVMs assign a value to each transliteration hypothesis $(h_i)$ using

$$g_{SVM}(h_i) = w \cdot x_i + b$$ (5)

where $w$ denotes a weight vector. Here, we use the predicted value of $g_{SVM}(h_i)$ rather than the predicted class of $h_i$ given by SVMs because our ranking function, as represented by Eq. (4), determines the relative ordering between $h_i$ and $h_j$ in $H$. A ranking function based on the maximum entropy model assigns a probability to $h_i$ using

$$g_{MEM}(h_i) = Pr(y_i = +1|x_i)$$ (6)

We can finally obtain a ranked list for the given $H$—the higher the $g(h_i)$ value, the better the $h_i$.

5.1 Features

We represent the feature vector, $x_i$, with two types of features. The first is the confidence scores of $h_i$ given by Eqs. (1)–(3) and the second is Web-based features — $Freq$, $DFreq_k$, and $PFreq_k$. To normalize $Freq$, $DFreq_k$, and $PFreq_k$, we use their relative frequency over $H$ as in Eqs. (7)–(9), where $k = 1, 2, 3$ and $l = 1, 2, 3$.

$$RF(h_i, W_l) = \frac{Freq(h_i, W_l)}{\sum_{h_j \in H} Freq(h_j, W_l)}$$ (7)

$$RDF_k(h_i, W_l) = \frac{DFreq_k(h_i, W_l)}{\sum_{h_j \in H} DFreq_k(h_j, W_l)}$$ (8)

$$RPF_k(h_i, W_l) = \frac{PFreq_k(h_i, W_l)}{\sum_{h_j \in H} PFreq_k(h_j, W_l)}$$ (9)

Figure 2 shows how to construct feature vector $x_i$ from a given English word, Rachel, and its Chinese hypotheses, $H$, generated from our transliteration systems. We can obtain $r$ Chinese transliteration hypotheses and classify them into positive and negative samples according to $y_i$. Note that $y_i = +1$ if and only if $h_i$ is registered as a counterpart of $S$ in the training data. The bottom of Fig. 2 shows our feature set representing $x_i$. There are three confidence scores in $P(h_i|S)$ according to transliteration models and the three Web-based features $Web(W_1)$, $Web(W_2)$, and $Web(W_3)$.

\(^3\)SVM\textsuperscript{light} (Joachims, 2002)

\(^4\)“Maximum Entropy Modeling Toolkit” (Zhang, 2004)
6 Experiments

We evaluated the effectiveness of our system in selecting CJK transliteration hypotheses. We used the same test set used in Li et al. (2004) (ECSet) for Chinese transliterations (Xinhua News Agency, 1992) and those used in Oh et al. (2006) for Japanese and Korean transliterations — EJSET and EKSET (Breen, 2003; Nam, 1997). We divided the test data into training, development, and blind test sets as in Table 2. The training set was used to train our three transliteration models to generate the n-best transliteration hypotheses. The development set was used to train hypothesis selection based on support vector machines and maximum entropy model.

We used the blind test set for evaluation. The evaluation was done in terms of word accuracy (WA). WA is the proportion of correct transliterations in the best hypothesis by a system to correct transliterations in the blind test set.

| System   | ECSet   | EJSet   | EKSet   |
|----------|---------|---------|---------|
| KANG00   | N/A     | N/A     | 54.1    |
| GOTO03   | N/A     | 54.3    | N/A     |
| LI04     | 70.1    | N/A     | N/A     |
| GM       | 69.0    | 61.6    | 59.0    |
| PM       | 56.6    | 54.4    | 56.7    |
| CM       | 69.9    | 65.0    | 65.1    |

Table 3: WA of individual transliteration systems (%)

6.1 Results: Web counts vs. Web mining

We compared our transliteration system with three previous ones, all of which were based on a grapheme-based model (Goto et al., 2003; Kang and Kim, 2000; Li et al., 2004). LI04 is an English-to-Chinese transliteration system, which simultaneously takes English and Chinese contexts into consideration (Li et al., 2004). KANG00 is an English-to-Korean transliteration system and GOTO03 is an English-to-Japanese one — they segment a chunk of English graphemes and identify the most relevant sequence of target graphemes corresponding to the chunk (Goto et al., 2003; Kang and Kim, 2000). GM, PM, and CM, which are respectively based on Eqs. (1)–(3), are the transliteration systems we used for generating transliteration hypotheses. Our transliteration systems showed comparable or better performance than the previous ones regardless of the language.

We compared simple Web counts with our Web mining for hypothesis selection. We used the same set of transliteration hypotheses \( \mathcal{H} \) then compared their performance in hypothesis selection with two measures, relative frequency and \( g(h_i) \). Tables 4 and 5 list the results. Here, “Upper bound” is a system that always selects the correct transliteration hypothesis if there is a correct one in \( \mathcal{H} \). “Upper bound” can

---

\(^{5}\)We set \( n = 10 \) for the n-best. Thus, \( n \leq r \leq 3 \times n \) where \( \mathcal{H} = \{h_1, h_2, \ldots, h_r\} \)

\(^{6}\)The WA of LI04 was taken from the literature, where the training data were the same as the union of our training set and the development set while the test data were the same as in our test set. In other words, LI04 used more training data than ours did. With the same setting as LI04, our GM, PM, and CM produced respective WAs of 70.0, 57.7, and 71.7.

\(^{7}\)We implemented KANG00 (Kang and Kim, 2000) and GOTO03 (Goto et al., 2003), and tested them with the same data as ours.
Table 4: Web counts (WC) vs. Web mining (WM): hypothesis selection by relative frequency (%)

| System | ECSet | EJSet | EKSet |
|--------|-------|-------|-------|
| WC     |       |       |       |
| MQ     | 16.1  | 40.4  | 34.7  |
| BSQ    | 45.8  | 74.0  | 72.4  |
| BBQ    | 34.9  | 78.1  | 79.3  |
| WM     |       |       |       |
| RF(W1) | 62.9  | 78.4  | 77.1  |
| RDF(W1)| 70.8  | 80.4  | 80.2  |
| RPF(W1)| 73.5  | 79.7  | 79.4  |
| RF(W2) | 63.5  | 76.2  | 74.8  |
| RDF(W2)| 67.1  | 79.2  | 78.9  |
| RPF(W2)| 69.6  | 79.1  | 78.4  |
| RF(W3) | 37.9  | 53.9  | 55.8  |
| RDF(W3)| 76.4  | 69.0  | 70.2  |
| RPF(W3)| 76.8  | 68.3  | 68.7  |
| Upper bound | 94.6 | 93.5 | 93.2 |

Table 5: Web counts (WC) vs. Web mining (WM): hypothesis selection by \( g(h_i) \) (%)

| System | ECSet | EJSet | EKSet |
|--------|-------|-------|-------|
| WC     |       |       |       |
| MEM    | 74.7  | 86.1  | 85.6  |
| SVM    | 74.8  | 86.9  | 86.5  |
| WM     |       |       |       |
| MEM    | 82.0  | 88.2  | 85.8  |
| SVM    | 83.9  | 88.5  | 86.7  |
| Upper bound | 94.6 | 93.5 | 93.2 |

The results in the tables show that our systems consistently outperformed systems based on Web counts, especially for Chinese. This was due to the difference between languages. Japanese and Chinese do not use spaces between words. However, Japanese is written using three different alphabet systems, called Hiragana, Katakana, and Kanji, that assist word segmentation. Moreover, words written in Katakana are usually Japanese transliterations of foreign words. This makes it possible for a Web search engine to effectively retrieve Web pages containing given Japanese transliterations. Like English, Korean has spaces between words (or word phrases). As the spaces in the languages reduce ambiguity in segmenting words, a Web search engine can correctly identify Web pages containing given Korean transliterations. In contrast, there is a severe word-segmentation problem with Chinese that causes Chinese Web search engines to incorrectly retrieve Web pages, as shown in Fig. 3. For example, Snippet1 is not related to “Aman” but to “a man”.

Figure 3: Snippets causing errors in Web counts
Snippets contain a super-string of a given Chinese query, which corresponds to “Academy” rather than to “Agard”, which is the English counterpart of the Chinese transliteration 阿加。Moreover, Web search engines ignore punctuation marks in Chinese. In Snippets and Snippets, “,” and “;” in the underlined terms are disregarded, so the Web counts based on such Web documents are noisy. Thus, noise in the Chinese Web counts causes systems based on Web counts to produce more errors than our systems do. Our proposed method can filter out such noise because our systems take punctuation marks and the contexts of transliterations in Web mining into consideration. Thus, our systems based on features mined from the Web were able to achieve the best performance. The results revealed that our systems based on the Web-mining technique can effectively be used to select transliteration hypotheses regardless of the language.

6.2 Contribution of Web corpora

|       | ECSet |       |       | EJSet |       |       | EKSet |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|       | SVM   | MEM   | SVM   | MEM   | SVM   | MEM   | SVM   | MEM   |
| Base  | 73.3  | 73.8  | 67.0  | 66.1  | 66.0  | 66.4  |       |       |
| W₁    | 81.7  | 79.7  | 87.6  | 87.3  | 86.1  | 85.1  |       |       |
| W₂    | 80.8  | 79.5  | 86.9  | 86.0  | 83.8  | 82.1  |       |       |
| W₃    | 77.2  | 76.7  | 83.0  | 82.8  | 79.8  | 77.3  |       |       |
| W₁+2  | 83.8  | 82.3  | 88.5  | 87.9  | 86.3  | 85.9  |       |       |
| W₁+3  | 81.9  | 80.1  | 87.6  | 87.8  | 86.1  | 84.7  |       |       |
| W₂+3  | 81.4  | 79.8  | 88.0  | 87.7  | 85.1  | 84.3  |       |       |
| W₄All | 83.9  | 82.0  | 88.5  | 88.2  | 86.7  | 85.8  |       |       |

Table 6: Contribution of Web corpora

In Web mining, we used W₁, W₂, and W₃, collected by respective queries Q₁=(S and hᵢ), Q₂=S, and Q₃=hᵢ. To investigate their contribution, we tested our proposed method with different combinations of Web corpora. “Base” is a baseline system that only uses Pr(hᵢ|S) as features but does not use features mined from the Web. We added features mined from different combinations of Web corpora to “Base” from W₁ to W₄All.

In Table 6, we can see that W₁, a set of Web pages retrieved by Q₁, tends to give more relevant information than W₂ and W₃, because Q₁ can search more Web pages containing both S and hᵢ in the top-100 snippets if S and hᵢ are a correct transliteration pair. Therefore, its performance tends to be superior in Table 6 if W₁ is used, especially for EKSet. However, as W₁ occasionally retrieves few snippets, it is not able to provide sufficient information. Using W₂ or W₃, we can address the problem. Thus, combinations of W₁ and others (W₁+2, W₁+3, W₄All) provided better WA than W₁.

7 Discussion

Several Web mining techniques for transliteration lexicons have been developed in the last few years (Jiang et al., 2007; Oh and Isahara, 2006). The main difference between ours and those previous ones is in the way a set of transliteration hypotheses (or candidates) is created.

Jiang et al. (2007) generated Chinese transliterations for given English words and searched the Web using the transliterations. They generated only the best transliteration hypothesis and focused on Web mining to select transliteration lexicons rather than selecting transliteration hypotheses. The best transliteration hypothesis was used to guide Web searches. Then, transliteration candidates were mined from the retrieved Web pages. Therefore, their performance greatly depended on their ability to mine transliteration candidates from the Web. However, this system might create errors if it cannot find a correct transliteration candidate from the retrieved Web pages. Because of this, their system’s coverage and WA were relatively poor than ours. However, our transliteration process was able to generate a set of transliteration hypotheses with excellent coverage and could thus achieve superior WA.

Oh and Isahara (2006) searched the Web using given source words and mined the retrieved Web pages to find target-language transliteration candidates. They extracted all possible sequences of target-language characters from the retrieved Web snippets as transliteration candidates for which the beginnings and endings of the given source word

8Since both Jiang et al.’s (2007) and ours used Chinese transliterations of personal names as a test set, we can indirectly compare our coverage and WA with theirs (Jiang et al., 2007). Jiang et al. (2007) achieved a 74.5% coverage of transliteration candidates and 47.5% WA, while ours achieved a 94.6% coverage of transliteration hypotheses and 82.0–83.9% WA.
and the extracted transliteration candidate were phonetically similar. However, while this can exponentially increase the number of transliteration candidates, ours used the n-best transliteration hypotheses but still achieved excellent coverage.

8 Conclusion

We have described a novel approach to selecting transliteration hypotheses based on Web mining. We first generated CJK transliteration hypotheses for a given English word and retrieved Web pages using the transliteration hypotheses and the given English word as queries for a Web search engine. We then mined features from the retrieved Web pages and trained machine-learning algorithms using the mined features. Finally, we selected transliteration hypotheses by ranking them. Our experiments revealed that our proposed method worked well regardless of the language, while simple Web counts were not effective, especially for Chinese.

Because our method was very effective in selecting transliteration pairs, we expect that it will also be useful for selecting translation pairs. We plan to extend our method in future work to selecting translation pairs.

References

Y. Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In Proc. of ACL ’02, pages 400–408.

J. Breen. 2003. EDICT Japanese/English dictionary .le. The Electronic Dictionary Research and Development Group, Monash University. http://www.csse.monash.edu.au/~jwb/edict.html.

I. Goto, N. Kato, N. Uratani, and T. Ebara. 2003. Transliteration considering context information based on the maximum entropy method. In Proc. of MT-Summit IX, pages 125–132.

Gregory Grefenstette, Yan Qu, and David A. Evans. 2004. Mining the Web to create a language model for mapping between English names and phrases and Japanese. In Proc. of Web Intelligence, pages 110–116.

Long Jiang, Ming Zhou, Lee-Feng Chien, and Cheng Niu. 2007. Named entity translation with Web mining and transliteration. In Proc. of IJCAI, pages 1629–1634.

Thorsten Joachims. 2002. Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms. Kluwer Academic Publishers.

I. H. Kang and G. C. Kim. 2000. English-to-Korean transliteration using multiple unbounded overlapping phoneme chunks. In Proc. of COLING ’00, pages 418–424.

Mirella Lapata and Frank Keller. 2005. Web-based models for natural language processing. ACM Trans. Speech Lang. Process., 2(1):3.

H. Li, M. Zhang, and J. Su. 2004. A joint source-channel model for machine transliteration. In Proc. of ACL ’04, pages 160–167.

H.M. Meng, Wai-Kit Lo, Berlin Chen, and K. Tang. 2001. Generating phonetic cognates to handle named entities in English-Chinese cross-language spoken document retrieval. In Proc. of Automatic Speech Recognition and Understanding, 2001. ASRU ’01, pages 311–314.

Y. S. Nam. 1997. Foreign dictionary. Sung An Dang.

Jong-Hoon Oh and Key-Sun Choi. 2002. An English-Korean transliteration model using pronunciation and contextual rules. In Proc. of COLING2002, pages 758–764.

Jong-Hoon Oh and Hitoshi Isahara. 2006. Mining the Web for transliteration lexicons: Joint-validation approach. In Web Intelligence, pages 254–261.

Jong-Hoon Oh, Key-Sun Choi, and Hitoshi Isahara. 2006. A comparison of different machine transliteration models. Journal of Artificial Intelligence Research (JAIR), 27:119–151.

Yan Qu and Gregory Grefenstette. 2004. Finding ideographic representations of Japanese names written in Latin script via language identification and corpus validation. In Proc. of ACL ’04, pages 183–190.

Richard Schwartz and Yen-Lu Chow. 1990. The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypothesis. In Proc. of ICASSP ’90, pages 81–84.

Xinhua News Agency. 1992. Chinese transliteration of foreign personal names. The Commercial Press.

L. Zhang. 2004. Maximum entropy modeling toolkit for python and C++. http://homepages.inf.ed.ac.uk/s0450736/software/maxent/manual.pdf.