# Direct Orthographical Mapping for Machine Transliteration

**ZHANG Min** **LI Haizhou** **SU Jian**

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 11961
{mzhang, hli, sujian}@i2r.a-star.edu.sg

## Abstract

Machine transliteration/*back*-transliteration plays an important role in many multilingual speech and language applications. In this paper, a novel framework for machine transliteration/*back*-transliteration that allows us to carry out direct orthographical mapping (DOM) between two different languages is presented. Under this framework, a joint source-channel transliteration model, also called *n*-gram transliteration model (*n*-gram TM), is further proposed to model the transliteration process. We evaluate the proposed methods through several transliteration/*back*-transliteration experiments for English/Chinese and English/Japanese language pairs. Our study reveals that the proposed method not only reduces an extensive system development effort but also improves the transliteration accuracy significantly.

## 1 Introduction

Many technical terms and proper names, such as personal, location and organization names, are translated from one language into another language with approximate phonetic equivalents. The phonetic translation from the native language to foreign language is defined as **transliteration**; conversely, the process of recalling a word in native language from a transliteration is defined as ***back*-transliteration**. For example, English name "Smith" and "史密斯 (*pinyin*[1]: Shi-Mi-Si)" in Chinese form a pair of transliteration and *back*-transliteration. In many natural language processing tasks, such as multilingual named entity and term processing, machine translation, corpus alignment, cross lingual information retrieval and automatic bilingual dictionary compilation, automatic name transliteration has become an indispensable component.

Recent efforts are reported for several language pairs, such as English/Chinese (Meng *et al.,* 2001; Virga *et al.*, 2003; Lee *et al.,* 2003; Gao *et al.*, 2004; Guo *et al.*, 2004), English/Japanese (Knight *et al.*, 1998; Brill *et al.,* 2001; Bilac *et al.*, 2004), English/Korean (Oh *et al.*, 2002; Sung *et al.,* 2000), and English/Arabic (Yaser *et al.*, 2002). Most of the reported works utilize a phonetic clue to resolve the transliteration through a multiple step phonemic mapping where algorithms, such as dictionary lookup, rule-based and machine learning-based approaches, have been well explored.

In this paper, we will discuss the limitation of the previous works and present a novel framework for machine transliteration. The new framework carries out the transliteration by direct orthographical mapping (DOM) without any intermediate phonemic mapping. Under this framework, we further propose a joint source-channel transliteration mode (*n*-gram TM) as an alternative machine learning-based approach to model the source-target word orthographic association. Without the loss of generality, we evaluate the performance of the proposed method for English/Chinese and English/Japanese pairs. An experiment that compares the proposed method with several state-of-art approaches is also presented. The results reveal that our method outperforms other previous methods significantly.

The reminder of the paper is organized as follows. Section 2 reviews the previous work. In section 3, the DOM framework and *n*-gram TM model are formulated. Section 4 describes the evaluation results and compares our method with other reported work. Finally, we conclude the study with some discussions.

## 2 Previous Work

The topic of machine transliteration has been studied extensively for several different language pairs, and many techniques have been proposed. To better understand the nature of the problem, we review the previous work from two different viewpoints: the transliteration framework and the transliteration model. The transliteration model is built to capture the knowledge of bilingual phonetic association and subsequently is applied to the transliteration process.

[1] *Pinyin* is the standard Romanization of Chinese.

## 2.1 Transliteration Framework

The phoneme-based approach has received remarkable attention in the previous works (Meng *et al.,* 2001; Virga *et al.*, 2003; Knight *et al.*, 1998; Oh *et al.*, 2002; Sung *et al.,* 2000; Yaser *et al.*, 2002; Lee *et al.,* 2003). In general, this approach includes the following three intermediate phonemic/orthographical mapping steps:

1) Conversion of a source language word into its phonemic representation (grapheme-to-phoneme conversion, or G2P);
2) Transformation of the source language phonemic representation to the target language phonemic representation;
3) Generation of target language orthography from its phonemic representation (phoneme-to-grapheme conversion, or P2G).

To achieve phonetic equivalent transliteration, phoneme-based approach has become the most popular approach. However, the success of phoneme-based approach is limited by the following constraints:

1) Grapheme-to-phoneme conversion, originated from text-to-speech (TTS) research, is far from perfect (The Onomastica Consortium, 1995), especially for the name of different language origins.
2) Cross-lingual phonemic mapping presents a great challenge due to phonemic divergence between some language pairs, such as Chinese/English, Japanese/English (Wan and Verspoor, 1998; Meng *et al.,* 2001).
3) The conversion of phoneme-to-grapheme introduces yet another level of imprecision, esp. for the ideographic language, such as Chinese. Virga and Khudanpur (2003) reported 8.3% absolute accuracy drops when converting from *Pinyin* to Chinese character.

The three error-prone steps as stated above lead to an inferior overall system performance. The complication of multiple steps and introduction of intermediate phonemes also incur high cost in system development when moving from one language pair to another, because we have to work on language specific ad-hoc phonic rules.

## 2.2 Transliteration Model

Transliteration model is a knowledge base to support the execution of transliteration strategy. To build the knowledge base, machine learning or rule-based algorithms are adopted in phoneme-based approach. For instance, noisy-channel model (NCM) (Virga *et al.*, 2003; Lee *et al.,* 2003), HMM (Sung *et al.,* 2000), decision tree (Kang *et al.*, 2000), transformation-based learning (Meng *et al.,* 2001), statistical machine transliteration model (Lee *et al.,* 2003), finite state transducers (Knight *et al.*, 1998) and rule-based approach (Wan *et al.*, 1998; Oh *et al.*, 2002). It is observed that the reported transliteration models share a common strategy, that is:

1) To model the transformation rules;
2) To model the target language;
3) To model the above both;

However, the modeling of different knowledge is always done independently. For example, NCM and HMM (Virga *et al.*, 2003; Lee *et al.,* 2003; Sung *et al.,* 2000) model the transformation mapping rules and the target language separately; decision tree (Kang *et al.*, 2000), transformation-based learning (Meng *et al.,* 2001), finite state transducers (Knight *et al.*, 1998) and statistical machine transliteration model (Lee *et al.,* 2003) only model the transformation rules.

## 3 Direct Orthographical Mapping

To overcome the limitation of phoneme-based approach, we propose a unified framework for machine transliteration, direct orthographical mapping (DOM). The DOM framework tries to model phonetic equivalent association by fully exploring the orthographical contextual information and the orthographical mapping. Under the DOM framework, we propose a joint source-channel transliteration model (*n*-gram TM) to capture the source-target word orthographical mapping relation and the contextual information. Unlike the noisy-channel model, the joint source-channel model does not try to capture how the source names can be mapped to the target names, but rather how both source and target names can be generated simultaneously.

The proposed framework is applicable to all language pairs. For simplicity, in this section, we take English/Chinese pair as example in the formulation, where *E2C* refers to English to Chinese transliteration and *C2E* refers to Chinese to English *back*-transliteration.

## 3.1 Transliteration Pair and Alignment

Suppose that we have an English name $\alpha = x_1 ... x_i ... x_m$ and a Chinese transliteration $\beta = y_1 ... y_i ... y_n$ where $x_i$ are English letters and $y_j$ are Chinese characters. The English name $\alpha$ and its Chinese Transliteration $\beta$ can be segmented into a series of substrings: $\alpha = e_1 e_2 ... e_K$ and $\beta = c_1 c_2 ... c_K$ ( $k < \min(m,n)$ ). We call the substring as transliteration unit and each English transliteration unit $e_i$ is aligned with a corresponding Chinese transliteration unit $c_i$ to form a transliteration pair. An alignment between $\alpha$ and $\beta$ is defined as $\gamma$ with

$$<e,c>_1 = <e_1, c_1>$$

$$<e,c>_2 = <e_2, c_2> \ \ldots$$

and $<e,c>_K = <e_K, c_K>$. A transliteration pair $<e,c>_i$ represents a two-way mapping between $e_i$ and $c_i$. A unit could be a Chinese character or a monograph, a digraph or a trigraph and so on for English. For example, "阿|a 布|b 鲁|ru 佐|zzo" is one alignment of Chinese-English word pair "阿布鲁佐" and "abruzzo".

### 3.2 DOM Transliteration Framework

By the definition of $\alpha$, $\beta$ and $\gamma$, the *E2C* transliteration can be formulated as

$$\begin{aligned} \overline{\beta} &= \arg\max_{\beta} P(\alpha,\beta) \\ &= \arg\max_{\beta} \sum_{\gamma} P(\alpha,\beta,\gamma) \\ &\approx \arg\max_{\beta}(\arg\max_{\gamma} P(\alpha,\beta,\gamma)) \\ &= \arg\max_{\beta,\gamma} P(\alpha,\beta,\gamma) \end{aligned} \qquad (1)$$

Similarly the *C2E back*-transliteration as

$$\overline{\alpha} \approx \arg\max_{\alpha,\gamma} P(\alpha,\beta,\gamma) \qquad (2)$$

To reduce the computational complexity, in *eqn.* (1), common practice is to replace the summation with maximization.

The *eqn.* (1) and (2) formulate the DOM transliteration framework. $P(\alpha,\beta,\gamma)$ is the joint probability of $\alpha$, $\beta$ and $\gamma$, whose definition depends on the transliteration model which will be discussed in the next two subsections. Unlike the phoneme-based approach, DOM does not need to explicitly model any phonetic information of either source or target language. Assuming sufficient training corpus, DOM transliteration framework is to capture the phonetic equivalents through orthographic mapping or transliteration pair $<e,c>_i$. By eliminating the potential imprecision introduced through a multiple-step phonetic mapping in the phoneme-based approach, DOM is expected to outperform. In contrast to phoneme-based approach, DOM is purely data-driven, therefore can be extended across different language pairs easily.

### 3.3 *n*-gram TM under DOM

Given $\alpha$ and $\beta$, the joint probability of $P(\alpha,\beta,\gamma)$ is the probability of alignment $\gamma$, which can be formulated as follows:

$$\begin{aligned} P(\alpha,\beta,\gamma) &= P(\alpha,\beta \mid \gamma) * P(\gamma) \\ &= \prod_{k=1}^{K} P(<e,c>_k \mid <e,c>_1^{k-1}) \end{aligned} \qquad (3)$$

In *eqn.* (3), the transliteration pair is used as the token to derive *n*-gram statistics, so we call the model as *n*-gram TM transliteration model.

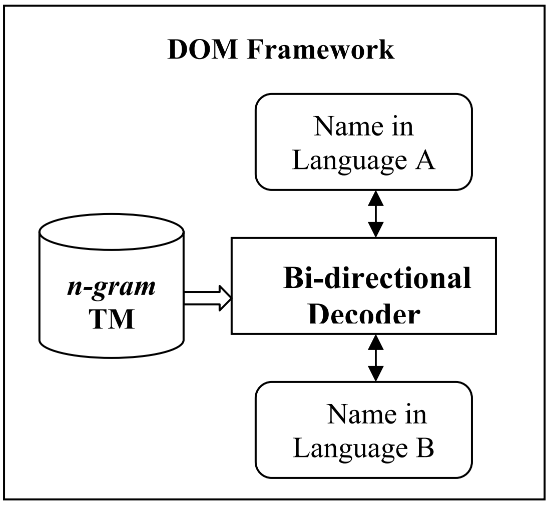

Figure 1. System structure of DOM

The above block diagram illustrates typical system structure of DOM. The training of *n*-gram TM model is discussed in section 3.5. Given a language pair, the bidirectional transliterations can be achieved with the same *n*-gram TM and using the same decoder.

### 3.4 DOM: *n*-gram TM *vs.* NCM

Noisy-channel model (NCM) has been well studied in the phoneme-based approach. Let's take *E2C* as an example to look into a bigram case to see what *n*-gram TM and NCM present to us under DOM. We have

$$\begin{aligned} P(\alpha,\beta,\gamma) &= P(\alpha,\gamma \mid \beta) * P(\beta) \\ &\approx \prod_{k=1}^{K} P(e_k \mid c_k) * P(c_k \mid c_{k-1}) \end{aligned} \qquad (4)$$

$$\begin{aligned} P(\alpha,\beta,\gamma) &= P(\alpha,\beta \mid \gamma) * P(\gamma) \\ &\approx \prod_{k=1}^{K} P(<e,c>_k \mid <e,c>_{k-1}) \end{aligned} \qquad (5)$$

where *eqn.* (4) and (5) are the bigram version of NCM and *n*-gram TM under DOM, respectively. The formulation of *eqn.* (4) could be interpreted as a HMM that has Chinese units as its hidden states and English transliteration units as the observations (Rabiner, 1989). Indeed, NCM consists of two models; one is the channel model or transliteration model, $\prod_{k=1}^{K} P(e_k \mid c_k)$, which tries to estimate the mapping probability between the two units;

another is the source model or language model, $\prod_{k=1}^{K} P(c_k \mid c_{k-1})$ , which tries to estimate the generative probability of the Chinese name, given the sequence of Chinese transliteration units. Unlike NCM, *n*-gram TM model does not try to capture how source names can be mapped into target names, but rather how source and target names can be generated simultaneously.

We can also study the two models from the contextual information usage viewpoint. One finds that *eqn*. (4) can be approximated by *eqn*. (5).

$$
\begin{aligned}
&P(<e,c>_k \mid <e,c>_{k-1}) \\
&\quad = P(e_k \mid c_k, <e,c>_{k-1}) * P(c_k \mid <e,c>_{k-1}) \quad (6) \\
&\quad \approx P(e_k \mid c_k) * P(c_k \mid c_{k-1})
\end{aligned}
$$

*Eqn*. (6) shows us that the context information $<e,c>_{k-1}$ and $e_{k-1}$ are absent in the channel model and source model of NCM, respectively. In this way, one could argue that *n*-gram TM model captures more context information than traditional NCM model. With adequate and sufficient training data, *n*-gram TM is expected to outperform NCM in the decoding.

### 3.5 Transliteration Alignment Training

For the *n*-gram TM model training, the bilingual name corpus needs to be aligned firstly at the transliteration unit level. The maximum likelihood approach, through EM algorithm (Dempster *et al.*, 1977) is employed to infer such an alignment.

The aligning process is different from that of transliteration given in *eqn*. (1) or (2), here we have a fixed bilingual entries, $\alpha$ and $\beta$ . The aligning process is just to find the alignment segmentation $\bar{\gamma}$ between the two strings that maximizes the joint probability:

$$\bar{\gamma} = \arg\max_{\gamma} P(\alpha, \beta, \gamma) \qquad (7)$$

Kneser-Ney smoothing algorithm (Chen *et al.*, 1998) is applied to smooth the probability distribution. NCM model training is carried out in the similar way to *n*-gram TM. The difference between the two models lies in *eqn* (4) and (5).

### 3.6 Decoding Issue

The decoder searches for the most probabilistic path of transliteration pairs, given the word in source language, by resolving different combinations of alignments. Rather than Viterbi algorithm, we use stack decoder (Schwartz *et al.*, 1990) to get *N*-best results for further processing or as output for other applications.

## 4 The Experiments

### 4.1 Testing Environments

We evaluate our method through several experiments for two language pairs: English/Chinese and English/Japanese.

For English/Chinese language pair, we use a database from the bilingual dictionary "Chinese Transliteration of Foreign Personal Names" (Xinhua, 1992). The database includes a collection of 37,694 unique English entries and their official Chinese transliteration. The listing includes personal names of English, French, and many other origins. The following results for this language pair are estimated by 13-fold cross validation for more accurate. We report two types of error rates: word error rate and character error rate. In word error rate, a word is considered correct only if an exact match happens between transliteration and the reference. The character error rate is the sum of deletion, insertion and substitution errors. Only the top choice in *N*-best results is used for character error rate reporting.

For English/Japanese language pair, we use the same database as that in the literature (Bilac *et al.*, 2004)[2]. The database includes 7,021 Japanese words in *katakana* together with their English translation extracted from the EDICT dictionary[3]. 714 tokens of these entries are withheld for evaluation. Only word error rate is reported for this language pair.

### 4.2 Modeling

The alignment is done fully automatically along with the *n*-gram TM training process.

| | |
|---|---|
| # close set bilingual entries (full data) | 37,694 |
| # unique Chinese transliteration (close) | 28,632 |
| # training entries for open test | 34,777 |
| # test entries for open test | 2,896 |
| # unique transliteration pairs $T$ | 5,640 |
| # total transliteration pairs $W_T$ | 119,364 |
| # unique English units $E$ | 3,683 |
| # unique Chinese units $C$ | 374 |
| # bigram TM $P(<e,c>_k \mid <e,c>_{k-1})$ | 38,655 |
| # NCM Chinese bigram $P(c_k \mid c_{k-1})$ | 12,742 |

Table 1. Modeling statistics (E-C)

Table 1 reports statistics in the model training for English/Chinese pair, and table 2 is for English/Japanese pair.

[2] We thank Mr. Slaven Bilac for letting us use his testing setup as a reference.

[3] ftp://ftp.cc.monash.edu.au/pub/nihongo/.

| | |
|---|---|
| # close set bilingual entries (full data) | 7,021 |
| # training entries for open test | 6,307 |
| # test entries for open test | 714 |
| # unique transliteration pairs $T$ | 2,173 |
| # total transliteration pairs $W_T$ | 28,366 |
| # unique English units $E$ | 1,216 |
| # unique Japanese units $J$ | 276 |
| # bigram TM $P(<e,j>_k \| <e,j>_{k-1})$ | 9,754 |

Table 2. Modeling statistics (E-J)

### 4.3 *E2C* Transliteration

In this experiment, we conduct both open and closed tests for *n*-gram TM and NCM models under DOM paradigm. Results are reported in Table 3 and Table 4.

| | open (word) | open (char) | Closed (word) | closed (char) |
|---|---|---|---|---|
| 1-gram | 45.6% | 21.1% | 44.8% | 20.4% |
| 2-gram | 31.6% | 13.6% | 10.8% | 4.7% |
| 3-gram | 29.9% | 10.8% | 1.6% | 0.8% |

Table 3. *E2C* error rates for *n*-gram TM tests.

| | open (word) | open (char) | closed (word) | closed (char) |
|---|---|---|---|---|
| 1-gram | 47.3% | 23.9% | 46.9% | 22.1% |
| 2-gram | 39.6% | 20.0% | 16.4% | 10.9% |
| 3-gram | 39.0% | 18.8% | 7.8% | 1.9% |

Table 4. *E2C* error rates for NCM tests

Not surprisingly, the result shows that *n*-gram TM, which benefits from the joint source-channel model coupling both source and target contextual information into the model, is superior to NCM in all the test cases.

### 4.4 *C2E Back*-Transliteration

The *C2E back*-transliteration is more challenging than *E2C* transliteration. Experiment results are reported in Table 5. As expected, *C2E* error rate is much higher than that of *E2C*.

| | open (word) | Open (letter) | closed (word) | closed (letter) |
|---|---|---|---|---|
| 1 gram | 82.3% | 28.2% | 81% | 27.7% |
| 2 gram | 63.8% | 20.1% | 40.4% | 12.3% |
| 3 gram | 62.1% | 19.6% | 14.7% | 5.0% |

Table 5. *C2E* error rate for *3*-gram TM tests

Table 6 reports the *N*-best word error rates for both *E2C* and *C2E* which implies the potential of error reduction by using secondary knowledge source, such as table looking-up. The *N*-best error rates are also reduced greatly at 10-best level.

| | *E2C* open | *E2C* closed | *C2E* open | *C2E* Closed |
|---|---|---|---|---|
| 1-best | 29.9% | 1.6% | 62.1% | 14.7% |
| 5-best | 8.2% | 0.94% | 43.3% | 5.2% |
| 10-best | 5.4% | 0.90% | 24.6% | 4.8% |

Table 6. *N*-best word error rates for *3*-gram TM

### 4.5 Discussions of DOM

Due to lack of standard data sets, the DOM framework is unable to make a straightforward comparison with other approaches. Nevertheless, we list some reported studies on other databases of *E2C* tasks in Table 7 and those of *C2E* tasks in Table 8 for reference purpose. In Table 7, the reference data are extracted from Table 1 and 3 of (Virga *et al.*, 2003), where only character and *Pinyin* error rates are reported. The first 4 setups by Virga *et al.* all adopted the phoneme-based approach. In table 8, the reference data are extracted from Table 2 and Figure 4 of (Guo *et al.*, 2004), where word error rates are reported.

| System | Training size | Test size | *Pinyin* errors | Char errors |
|---|---|---|---|---|
| Meng *et al.* | 2,233 | 1,541 | 52.5% | N/A |
| Small MT | 2,233 | 1,541 | 50.8% | 57.4% |
| Big MT | 3,625 | 250 | 49.1% | 57.4% |
| Huge MT (Big MT) | 309,019 | 3,122 | 42.5% | N/A |
| *3*-gram TM/DOM | 34,777 | 2,896 | <10.8% | 10.8% |
| *3*-gram NCM/DOM | 34,777 | 2,896 | <18.8% | 18.8% |

Table 7. Performance Comparison of *E2C*

Since we have obtained results in character already and the character to *Pinyin* mapping is one-to-one in the 374 legitimate Chinese characters for transliteration in our implementation, we expect less *Pinyin* error than character error in Table 7.

| | Training size | Test size | 1-best | 10-best |
|---|---|---|---|---|
| Guo *et al.* | 424,788 | 500 | >82.0% | >50.0% |
| *3*-gram TM/DOM | 34,777 | 2,896 | 62.1% | 24.6% |

Table 8. Performance Comparison of *C2E*

For *E2C*, Table 7 shows that even with an 8 times larger database than ours, Huge MT (Big

MT) test case who reports the best performance still generates 3 times *Pinyin* error rate than ours. For *C2E*, Table 8 shows that even with only 9 percent training set, our approach can still make 20 percent absolute word error rate reduction. Thus, although the experiment are done in different environments, to some extend, Table 7 and Table 8 reveal that the *n*-gram TM/DOM outperforms other techniques for the case of English/Chinese transliteration/*back*-transliteration significantly.

### 4.6 English/Japanese Transliteration

In this experiment, we conduct both open and closed tests for *n*-gram TM on English/Japanese transliteration and *back*-transliteration. We use the same training and testing setups as those in (Bilac *et al.*, 2004).

Table 9 reports the results from three different transliteration mechanisms. Case 1 is the *3*-gram TM under DOM; Case 2 is Case 1 integrated with a dictionary lookup validation process during decoding; Case 3 is extracted from (Bilac *et al.*, 2004). Similar to English/Chinese transliteration, one can find that *J2E back*-transliteration is more challenging than *E2J* transliteration in both open and closed cases. It is also found that word error rates are reduced greatly at 10-best level.

(Bilac *et al.*, 2004) proposed a hybrid-method of grapheme-based and phoneme-based for *J2E back*-transliteration, where the whole EDICT dictionary, including the test set, is used to train a LM. A LM unit is a word itself. In this way, the dictionary is used as a lookup table in the decoding process to help identify a valid choice among candidates. To establish comparison, we also integrate the dictionary lookup processing with the decoder, which is referred as Case 2 in Table 9. It is found that Case 2 presents a error reduction of 43.8%=(14.6-8.2)/14.6% for word over to those reported in (Bilac *et al.*, 2004). Furthermore, the *n*-gram TM/DOM approach is rather straightforward in implementation where direct orthographical mapping could potentially handle Japanese transliteration of names of different language origins, while the issues with non-English terms are reported in (Bilac *et al.*, 2004).

The DOM framework shows us a great improvement in performance with *n*-gram TM being the most successful implementation. Nevertheless, NCM presents another successful implementation of DOM framework. The *n*-gram TM and NCM under direct orthographic mapping (DOM) paradigm simplify the process and reduce the chances of conversion errors. The experiments also show that even with much less training data, DOM are still much more superior performance than the state of art solutions.

## 5 Conclusions

In this paper, we propose a new framework, direct orthographical mapping (DOM) for machine transliteration and *back*-transliteration. Under the DOM framework, we further propose a joint source-channel transliteration model, also called *n*-gram TM. We also implement the NCM model under DOM for reference. We use EM algorithm as an unsupervised training approach to train the *n*-gram TM and NCM. The proposed methods are tested on an English-Chinese name corpus and English-Japanese *katakana* word pair extracted from EDICT dictionary. The data-driven and one-step mapping strategies greatly reduce the development efforts of machine transliteration systems and improve accuracy significantly over earlier reported results. We also find the *back*-transliteration is more challenging than the transliteration.

The DOM framework demonstrates several unique edges over phoneme-based approach:

| | | English-Japanese Transliteration | | Japanese-English Back-transliteration | |
|---|---|---|---|---|---|
| | | open test | closed test | open test | closed test |
| Case 1*:* *3*-gram TM/DOM | 1-best | 40.5% | 13.5% | 62.8% | 17.9% |
| | 10-best | 13.2% | 0.8% | 17.9% | 2.1% |
| Case 2: *3*-gram TM/DOM with dictionary lookup | 1-best | 5.4% | 0.7% | 8.2% | 1.2% |
| | 10-best | 0.7% | 0% | 1.7% | 0.3% |
| Case 3: Bilac *et al.*, 2004 | 1-best | N/A | N/A | 14.6% | N/A |
| | 10-best | N/A | N/A | 2.2% | N/A |

Table 9. Experiment results of English-Japanese Transliteration

1) By skipping the intermediate phonemic interpretation, the transliteration error rate is reduced significantly;
2) Transliteration models under DOM are data-driven. Assuming sufficient training corpus, the modeling approach applies to different language pairs;
3) DOM presents a paradigm shift for machine transliteration, that provides a platform for implementation of many other transliteration models;

The *n*-gram TM is a successful implementation of DOM framework due to the following aspects:

1) *N*-gram TM captures contextual information in both source and target languages jointly; unlike the phoneme-based approach, the modeling of transformation rules and target language is tightly coupled in *n*-gram TM model.
2) As *n*-gram TM uses transliteration pair as modeling unit, the same model applies to bi-directional transliteration;
3) The bilingual aligning process is integrated into the decoding process in *n*-gram TM, which allows us to achieve a joint optimization of alignment and transliteration automatically. Hence manual pre-alignment is unnecessary.

Named entities are sometimes translated in combination of transliteration and meanings. As the proposed framework allows direct orthographical mapping, we are extending our approach to handle such name translation. We also extending our method to handle the disorder and fertility issues in named entity translation.

## References

Chun-Jen Lee and Jason S. Chang, 2003. *Acquisition of English-Chinese Transliteration Word Pairs from Parallel-Aligned Texts using a Statistical Machine Translation Model*, Proceedings of HLT-NAACL Workshop: Building and Using parallel Texts Data Driven Machine Translation and Beyond, 2003, Edmonton, pp. 96-103

Dempster, A.P., N.M. Laird and D.B.Rubin, 1977. *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Stat. Soc., Ser. B. Vol. 39

Eric Brill, Garry Kacmarcik and Chris Brockrtt, 2001. *Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs.* Proceeding of NLPRS'01

Helen M. Meng, Wai-Kit Lo, Berlin Chen and Karen Tang. 2001. *Generate Phonetic Cognates to Handle Name Entities in English-Chinese cross-language spoken document retrieval*, Proceedings of ASRU 2001

Jong-Hoon Oh and Key-Sun Choi, 2002. *An English-Korean Transliteration Model Using Pronunciation and Contextual Rules*, Proceedings of COLING 2000

Kang B.J. and Key-Sun Choi, 2000. *Automatic Transliteration and Back-transliteration by Decision Tree Learning*, Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, Greece

K. Knight and J. Graehl. 1998. *Machine Transliteration*, Computational Linguistics, Vol 24, No. 4

Paola Virga, Sanjeev Khudanpur, 2003. *Transliteration of Proper Names in Cross-lingual Information Retrieval*. Proceedings of ACL 2003 workshop MLNER, 2003

Rabiner, Lawrence R. 1989, *A tutorial on hidden Markov models and selected applications in speech recognition*, Proceedings of the IEEE 77(2)

Schwartz, R. and Chow Y. L., 1990. *The N-best algorithm: An efficient and Exact procedure for finding the N most likely sentence hypothesis*, Proceedings of ICASSP 1990, Albuquerque, pp. 81-84

Slaven Bilac and Hozumi Tanaka, 2004. *Improving Back-Transliteration by Combining Information Sources*. Proceedings of IJCNLP-04, Haian, pp. 542-547

Stephen Wan and Cornelia Maria Verspoor, 1998. *Automatic English-Chinese name transliteration for development of multilingual resources*. Proceedings of COLING-ACL'98

Stanley F. Chen and Joshua Goodman, 1998. *An Empirical Study of Smoothing Techniques for Language Modeling*, TR-10-98, Computer Science Group, Harvard Universituy. 1998

Sung Young Jung, Sung Lim Hong and Eunok Paek, 2000. *An English to Korean Transliteration Model of Extended Markov Window*, Proceedings of COLING 2000

The Onomastica Consortium, 1995. *The Onomastica interlanguage pronunciation lexicon*, Proceedings of EuroSpeech, Madrid, Spain, pp829-832

Wei Gao, Kam-Fai Wong and Wai Lam, 2004. *Phoneme-based Transliteration of Foreign Names for OOV Problems*. Proceedings of IJCLNP-04, Hainan, pp 374-381

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

Yaser Al-Onaizan and Kevin Knight, 2002. *Translating named entities using monolingual and bilingual resources*. Proceedings of the 40th ACL, Philadelphia, 2002, pp. 400-408

Yuqing Guo and Haifeng Wang, 2004. Chinese-to-English Backward Machine Transliteration. Companion Volume to the Proceedings of IJCNLP-04, Hainan, pp 17-20