A Joint Source-Channel Model for Machine Transliteration

Li Haizhou, Zhang Min, Su Jian

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{hli,sujian,mzhang}@i2r.a-star.edu.sg

Abstract

Most foreign names are transliterated into Chinese, Japanese or Korean with approximate phonetic equivalents. The transliteration is usually achieved through intermediate phonemic mapping. This paper presents a new framework that allows direct orthographical mapping (DOM) between two different languages, through a joint source-channel model, also called *n*-gram transliteration model (TM). With the *n*-gram TM model, we automate the orthographic alignment process to derive the aligned transliteration units from a bilingual dictionary. The *n*-gram TM under the DOM framework greatly reduces system development effort and provides a quantum leap in improvement in transliteration accuracy over that of other state-of-the-art machine learning algorithms. The modeling framework is validated through several experiments for English-Chinese language pair.

1 Introduction

In applications such as cross-lingual information retrieval (CLIR) and machine translation, there is an increasing need to translate out-of-vocabulary words from one language to another, especially from alphabet language to Chinese, Japanese or Korean. Proper names of English, French, German, Russian, Spanish and Arabic origins constitute a good portion of out-of-vocabulary words. They are translated through transliteration, the method of translating into another language by preserving how words sound in their original languages. For writing foreign names in Chinese, transliteration always follows the original romanization. Therefore, any foreign name will have only one *Pinyin* (romanization of Chinese) and thus in Chinese characters.

In this paper, we focus on automatic Chinese transliteration of foreign alphabet names. Because some alphabet writing systems use various diacritical marks, we find it more practical to write names containing such diacriticals as they are rendered in English. Therefore, we refer all foreign-Chinese transliteration to English-Chinese transliteration, or *E2C*.

Transliterating English names into Chinese is not straightforward. However, recalling the original from Chinese transliteration is even more challenging as the *E2C* transliteration may have lost some original phonemic evidences. The Chinese-English backward transliteration process is also called *back*-transliteration, or *C2E* (Knight & Graehl, 1998).

In machine transliteration, the noisy channel model (NCM), based on a phoneme-based approach, has recently received considerable attention (Meng et al. 2001; Jung et al, 2000; Virga & Khudanpur, 2003; Knight & Graehl, 1998). In this paper we discuss the limitations of such an approach and address its problems by *firstly* proposing a paradigm that allows direct orthographic mapping (DOM), *secondly* further proposing a joint source-channel model as a realization of DOM. Two other machine learning techniques, NCM and ID3 (Quinlan, 1993) decision tree, also are implemented under DOM as reference to compare with the proposed *n*-gram TM.

This paper is organized as follows: In section 2, we present the transliteration problems. In section 3, a joint source-channel model is formulated. In section 4, several experiments are carried out to study different aspects of proposed algorithm. In section 5, we relate our algorithms to other reported work. Finally, we conclude the study with some discussions.

2 Problems in transliteration

Transliteration is a process that takes a character string in source language as input and generates a character string in the target language as output. The process can be seen conceptually as two levels of decoding: segmentation of the source string into transliteration units; and relating the source language transliteration units with units in the target language, by resolving different combinations of alignments and unit mappings. A

unit could be a Chinese character or a monograph, a digraph or a trigraph and so on for English.

2.1 Phoneme-based approach

The problems of English-Chinese transliteration have been studied extensively in the paradigm of noisy channel model (NCM). For a given English name *E* as the observed channel output, one seeks *a posteriori* the most likely Chinese transliteration *C* that maximizes *P(C|E)*. Applying Bayes rule, it means to find *C* to maximize

$$P(E,C) = P(E \mid C)*P(C) \qquad (1)$$

with equivalent effect. To do so, we are left with modeling two probability distributions: *P(E|C)*, the probability of transliterating *C* to *E* through a noisy channel, which is also called transformation rules, and *P(C)*, the probability distribution of source, which reflects what is considered good Chinese transliteration in general. Likewise, in *C2E back*-transliteration, we would find *E* that maximizes

$$P(E,C) = P(C \mid E)*P(E) \qquad (2)$$

for a given Chinese name.

In *eqn* (1) and (2), *P(C)* and *P(E)* are usually estimated using *n*-gram language models (Jelinek, 1991). Inspired by research results of grapheme-to-phoneme research in speech synthesis literature, many have suggested phoneme-based approaches to resolving *P(E|C)* and *P(C|E),* which approximates the probability distribution by introducing a phonemic representation. In this way, we convert the names in the source language, say *E,* into an intermediate phonemic representation *P*, and then convert the phonemic representation into the target language, say Chinese *C*. In *E2C* transliteration, the phoneme-based approach can be formulated as *P(C|E) = P(C|P)P(P|E)* and conversely we have *P(E|C) = P(E|P)P(P|C)* for *C2E back*-transliteration.

Several phoneme-based techniques have been proposed in the recent past for machine transliteration using transformation-based learning algorithm (Meng et al. 2001; Jung et al, 2000; Virga & Khudanpur, 2003) and using finite state transducer that implements transformation rules (Knight & Graehl, 1998), where both handcrafted and data-driven transformation rules have been studied.

However, the phoneme-based approaches are limited by two major constraints, which could compromise transliterating precision, especially in English-Chinese transliteration:

1) Latin-alphabet foreign names are of different origins. For instance, French has different phonic rules from those of English. The phoneme-based approach requires derivation of proper phonemic representation for names of different origins. One may need to prepare multiple language-dependent grapheme-to-phoneme (G2P) conversion systems accordingly, and that is not easy to achieve (The Onomastica Consortium, 1995). For example, /Lafontant/ is transliterated into 拉丰唐(La-Feng-*Tang*) while /Constant/ becomes 康斯坦特(Kang-Si-*Tan*-*Te*)， where syllable /-tant/ in the two names are transliterated differently depending on the names' language of origin.

2) Suppose that language dependent grapheme-to-phoneme systems are attainable, obtaining Chinese orthography will need two further steps: a) conversion from generic phonemic representation to Chinese *Pinyin*; b) conversion from *Pinyin* to Chinese characters. Each step introduces a level of imprecision. Virga and Khudanpur (2003) reported 8.3% absolute accuracy drops when converting from *Pinyin* to Chinese characters, due to homophone confusion. Unlike Japanese *katakana* or Korean alphabet, Chinese characters are more ideographic than phonetic. To arrive at an appropriate Chinese transliteration, one cannot rely solely on the intermediate phonemic representation.

2.2 Useful orthographic context

To illustrate the importance of contextual information in transliteration, let's take name /Minahan/ as an example, the correct segmentation should be /Mi-na-han/, to be transliterated as 米−纳−汉 (*Pinyin*: Mi-Na-Han).

English	/mi-	-na-	-han/
Chinese	米	纳	汉
Pinyin	Mi	Nan	Han

However, a possible segmentation /Min-ah-an/ could lead to an undesirable syllabication of 明−阿−安 (*Pinyin*: Min-A-An).

English	/min-	-ah-	-an/
Chinese	明	阿	安
Pinyin	Min	A	An

According to the transliteration guidelines, a wise segmentation can be reached only after exploring the combination of the left and right context of transliteration units. From the computational point of view, this strongly suggests using a contextual *n*-gram as the knowledge base for the alignment decision.

Another example will show us how one-to-many mappings could be resolved by context. Let's take another name /Smith/ as an example. Although we

can arrive at an obvious segmentation /s-mi-th/, there are three Chinese characters for each of /s-/, /-mi-/ and /-th/. Furthermore, /s-/ and /-th/ correspond to overlapping characters as well, as shown next.

English	/s-	-mi-	-th/
Chinese 1	***史***	米	***斯***
Chinese 2	斯	***密***	史
Chinese 3	思	麦	瑟

A human translator will use transliteration rules between English syllable sequence and Chinese character sequence to obtain the best mapping *史-密-斯*, as indicated in italic in the table above.

To address the issues in transliteration, we propose a direct orthographic mapping (DOM) framework through a joint source-channel model by fully exploring orthographic contextual information, aiming at alleviating the imprecision introduced by the multiple-step phoneme-based approach.

3 Joint source-channel model

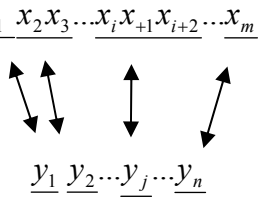
In view of the close coupling of the source and target transliteration units, we propose to estimate *P(E,C)* by a joint source-channel model, or *n*-gram transliteration model (TM). For *K* aligned transliteration units, we have

$$\begin{aligned} P(E,C) &= P(e_1,e_2...e_K,c_1,c_2...c_K) \\ &= P(<e,c>_1,<e,c>_2 \ ...<e,c>_K) \\ &= \prod_{k=1}^{K} P(<e,c>_k|<e,c>_1^{k-1}) \end{aligned} \quad (3)$$

which provides an alternative to the phoneme-based approach for resolving *eqn*. (1) and (2) by eliminating the intermediate phonemic representation.

Unlike the noisy-channel model, the joint source-channel model does not try to capture how source names can be mapped to target names, but rather how source and target names can be generated simultaneously. In other words, we estimate a joint probability model that can be easily marginalized in order to yield conditional probability models for both transliteration and *back*-transliteration.

Suppose that we have an English name $\alpha = x_1 x_2 ... x_m$ and a Chinese transliteration $\beta = y_1 y_2 ... y_n$ where x_i are letters and y_j are Chinese characters. Oftentimes, the number of letters is different from the number of Chinese characters. A Chinese character may correspond to a letter substring in English or vice versa.



where there exists an alignment γ with

$$<e,c>_1 = <x_1, y_1>$$
$$<e,c>_2 = <x_2x_3, y_2> \ \ldots$$

and $<e,c>_K = <x_m, y_n>$. A transliteration unit correspondence $<e,c>$ is called a transliteration pair. Then, the *E2C* transliteration can be formulated as

$$\overline{\beta} = \arg\max_{\beta,\gamma} P(\alpha,\beta,\gamma) \quad (4)$$

and similarly the *C2E back*-transliteration as

$$\overline{\alpha} = \arg\max_{\alpha,\gamma} P(\alpha,\beta,\gamma) \quad (5)$$

An *n*-gram transliteration model is defined as the conditional probability, or transliteration probability, of a transliteration pair $<e,c>_k$ depending on its immediate *n* predecessor pairs:

$$\begin{aligned} P(E,C) &= P(\alpha,\beta,\gamma) \\ &= \prod_{k=1}^{K} P(<e,c>_k|<e,c>_{k-n+1}^{k-1}) \end{aligned} \quad (6)$$

3.1 Transliteration alignment

A bilingual dictionary contains entries mapping English names to their respective Chinese transliterations. Like many other solutions in computational linguistics, it is possible to automatically analyze the bilingual dictionary to acquire knowledge in order to map new English names to Chinese and vice versa. Based on the transliteration formulation above, a transliteration model can be built with transliteration unit's *n*-gram statistics. To obtain the statistics, the bilingual dictionary needs to be aligned. The maximum likelihood approach, through EM algorithm (Dempster, 1977), allows us to infer

such an alignment easily as described in the table below.

The Expectation-Maximization algorithm

1. Bootstrap initial random alignment
2. Expectation: Update *n*-gram statistics to estimate probability distribution
3. Maximization: Apply the *n*-gram TM to obtain new alignment
4. Go to step 2 until the alignment converges
5. Derive a list transliteration units from final alignment as transliteration table

The aligning process is different from that of transliteration given in *eqn*. (4) or (5) in that, here we have fixed bilingual entries, α and β. The aligning process is just to find the alignment segmentation $\bar{\gamma}$ between the two strings that maximizes the joint probability:

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

A set of transliteration pairs that is derived from the aligning process forms a transliteration table, which is in turn used in the transliteration decoding. As the decoder is bounded by this table, it is important to make sure that the training database covers as much as possible the potential transliteration patterns. Here are some examples of resulting alignment pairs.

斯|s	尔|l	特|t	德|d
克|k	布|b	格|g	尔|r
尔|ll	克|c	罗|ro	里|ri
曼|man	姆|m	普|p	德|de
拉|ra	尔|le	阿|a	伯|ber
拉|la	森|son	顿|ton	特|tt
雷|re	科|co	奥|o	埃|e
马|ma	利|ley	利|li	默|mer

Knowing that the training data set will never be sufficient for every *n*-gram unit, different smoothing approaches are applied, for example, by using backoff or class-based models, which can be found in statistical language modeling literatures (Jelinek, 1991).

3.2 DOM: *n*-gram TM *vs.* NCM

Although in the literature, most noisy channel models (NCM) are studied under phoneme-based paradigm for machine transliteration, NCM can also be realized under direct orthographic mapping (DOM). Next, let's look into a bigram case to see what *n*-gram TM and NCM present to us. For *E2C* conversion, re-writing *eqn* (1) and *eqn* (6) , we have

$$P(\alpha, \beta, \gamma) \approx \prod_{k=1}^{K} P(e_k \mid c_k) P(c_k \mid c_{k-1}) \quad (8)$$

$$P(\alpha, \beta, \gamma) \approx \prod_{k=1}^{K} P(< e, c >_k \mid < e, c >_{k-1}) \quad (9)$$

The formulation of *eqn*. (8) could be interpreted as a hidden Markov model with Chinese characters as its hidden states and English transliteration units as the observations (Rabiner, 1989). The number of parameters in the bigram TM is potentially T^2, while in the noisy channel model (NCM) it's $T + C^2$, where T is the number of transliteration pairs and C is the number of Chinese transliteration units. In *eqn.* (9), the current transliteration depends on both Chinese and English transliteration history while in *eqn*. (8), it depends only on the previous Chinese unit.

As $T^2 >> T + C^2$, an *n*-gram TM gives a finer description than that of NCM. The actual size of models largely depends on the availability of training data. In Table 1, one can get an idea of how they unfold in a real scenario. With adequately sufficient training data, *n*-gram TM is expected to outperform NCM in the decoding. A perplexity study in section 4.1 will look at the model from another perspective.

4 The experiments[1]

We use a database from the bilingual dictionary "Chinese Transliteration of Foreign Personal Names" which was edited by Xinhua News Agency and was considered the *de facto* standard of personal name transliteration in today's Chinese press. The database includes a collection of 37,694 unique English entries and their official Chinese transliteration. The listing includes personal names of English, French, Spanish, German, Arabic, Russian and many other origins.

The database is initially randomly distributed into 13 subsets. In the open test, one subset is withheld for testing while the remaining 12 subsets are used as the training materials. This process is repeated 13 times to yield an average result, which is called the 13-fold open test. After experiments, we found that each of the 13-fold open tests gave consistent error rates with less than 1% deviation. Therefore, for simplicity, we randomly select one of the 13 subsets, which consists of 2896 entries, as the standard open test set to report results. In the close test, all data entries are used for training and testing.

[1] demo at http://nlp.i2r.a-star.edu.sg/demo.htm

4.1 Modeling

The alignment of transliteration units is done fully automatically along with the *n*-gram TM training process. To model the boundary effects, we introduce two extra units <s> and </s> for start and end of each name in both languages. The EM iteration converges at 8th round when no further alignment changes are reported. Next are some statistics as a result of the model training:

# 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

The most common metric for evaluating an *n*-gram model is the probability that the model assigns to test data, or perplexity (Jelinek, 1991). For a test set W composed of V names, where each name has been aligned into a sequence of transliteration pair tokens, we can calculate the probability of test set $p(W) = \prod_{v=1}^{V} P(\alpha_v, \beta_v, \gamma_v)$ by applying the *n*-gram models to the token sequence. The cross-entropy $H_p(W)$ of a model on data W is defined as $H_p(W) = -\frac{1}{W_T}\log_2 p(W)$ where W_T is the total number of aligned transliteration pair tokens in the data W. The perplexity $PP_p(W)$ of a model is the reciprocal of the average probability assigned by the model to each aligned pair in the test set W as $PP_p(W) = 2^{H_p(W)}$.

Clearly, lower perplexity means that the model describes better the data. It is easy to understand that closed test always gives lower perplexity than open test.

	TM open	NCM open	TM closed	NCM closed
1-gram	670	729	655	716
2-gram	324	512	151	210
3-gram	306	487	68	127

Table 2. Perplexity study of bilingual database

We have the perplexity reported in Table 2 on the aligned bilingual dictionary, a database of 119,364 aligned tokens. The NCM perplexity is computed using *n*-gram equivalents of *eqn.* (8) for *E2C* transliteration, while TM perplexity is based on those of *eqn* (9) which applies to both *E2C* and *C2E*. It is shown that TM consistently gives lower perplexity than NCM in open and closed tests. We have good reason to expect TM to provide better transliteration results which we expect to be confirmed later in the experiments.

The Viterbi algorithm produces the best sequence by maximizing the overall probability, $P(\alpha, \beta, \gamma)$. In CLIR or multilingual corpus alignment (Virga and Khudanpur, 2003), *N*-best results will be very helpful to increase chances of correct hits. In this paper, we adopted an *N*-best stack decoder (Schwartz and Chow, 1990) in both TM and NCM experiments to search for *N*-best results. The algorithm also allows us to apply higher order *n*-gram such as trigram in the search.

4.2 *E2C* transliteration

In this experiment, we conduct both open and closed tests for 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 *n*-gram NCM tests

In word error report, 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 error rate reporting. Not surprisingly, one can see 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.

4.3 *C2E back*-transliteration

The *C2E back*-transliteration is more challenging than *E2C* transliteration. Not many studies have been reported in this area. It is common that multiple English names are mapped into the same Chinese transliteration. In Table 1, we see only 28,632 unique Chinese transliterations exist for 37,694 English entries, meaning that some phonemic evidence is lost in the process of transliteration. To better understand the task, let's compare the complexity of the two languages presented in the bilingual dictionary.

Table 1 also shows that the 5,640 transliteration pairs are cross mappings between 3,683 English and 374 Chinese units. In order words, on average, for each English unit, we have 1.53 = 5,640/3,683 Chinese correspondences. In contrast, for each Chinese unit, we have 15.1 = 5,640/374 English *back*-transliteration units! Confusion is increased tenfold going backward.

The difficulty of *back*-transliteration is also reflected by the perplexity of the languages as in Table 5. Based on the same alignment tokenization, we estimate the monolingual language perplexity for Chinese and English independently using the *n*-gram language models $P(c_k \mid c_{k-n+1}^{k-1})$ and $P(e_k \mid e_{k-n+1}^{k-1})$. Without surprise, Chinese names have much lower perplexity than English names thanks to fewer Chinese units. This contributes to the success of *E2C* but presents a great challenge to *C2E back*-transliteration.

	1-gram	2-gram	3-gram
Chinese	207/206	97/86	79/45
English	710/706	265/152	234/67

Table 5 language perplexity comparison (open/closed test)

	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 6. *C2E* error rate for *n*-gram TM tests

	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 7. *N*-best word error rates for *3*-gram TM tests

A *back*-transliteration is considered correct if it falls within the multiple valid orthographically correct options. Experiment results are reported in Table 6. As expected, *C2E* error rate is much higher than that of *E2C*.

In this paper, the *n*-gram TM model serves as the sole knowledge source for transliteration. However, if secondary knowledge, such as a lookup table of valid target transliterations, is available, it can help reduce error rate by discarding invalid transliterations top-down the *N* choices. In Table 7, the word error rates for both *E2C* and *C2E* are reported which imply potential error reduction by secondary knowledge source. The *N*-best error rates are reduced significantly at *10*-best level as reported in Table 7.

5 Discussions

It would be interesting to relate *n*-gram TM to other related framework.

5.1 DOM: *n*-gram TM *vs.* ID3

In section 4, one observes that contextual information in both source and target languages is essential. To capture them in the modeling, one could think of decision tree, another popular machine learning approach. Under the DOM framework, here is the first attempt to apply decision tree in *E2C* and *C2E* transliteration.

With the decision tree, given a fixed size learning vector, we used top-down induction trees to predict the corresponding output. Here we implement ID3 (Quinlan, 1993) algorithm to construct the decision tree which contains questions and return values at terminal nodes. Similar to *n*-gram TM, for unseen names in open test, ID3 has backoff smoothing, which lies on the default case which returns the most probable value as its best guess for a partial tree path according to the learning set.

In the case of *E2C* transliteration, we form a learning vector of 6 attributes by combining 2 left and 2 right letters around the letter of focus e_k and 1 previous Chinese unit c_{k-1}. The process is illustrated in Table 8, where both English and Chinese contexts are used to infer a Chinese

character. Similarly, 4 attributes combining 1 left, 1 centre and 1 right Chinese character and 1 previous English unit are used for the learning vector in *C2E* test. An aligned bilingual dictionary is needed to build the decision tree.

To minimize the effects from alignment variation, we use the same alignment results from section 4. Two trees are built for two directions, *E2C* and *C2E*. The results are compared with those 3-gram TM in Table 9.

e_{k-2}	e_{k-1}	e_k	e_{k+1}	e_{k+2}	c_{k-1}		c_k
_	_	N	I	C	_	>	尼
_	N	I	C	E	尼	>	_
N	I	C	E	_	_	>	斯
I	C	E	_	_	斯	>	_

Table 8. *E2C* transliteration using ID3 decision tree for transliterating Nice to 尼斯 (尼|NI 斯|CE)

	open	closed
ID3 *E2C*	39.1%	9.7%
3-gram TM *E2C*	29.9%	1.6%
ID3 *C2E*	63.3%	38.4%
3-gram TM *C2E*	62.1%	14.7%

Table 9. Word error rate ID3 vs. *3*-gram TM

One observes that *n*-gram TM consistently outperforms ID3 decision tree in all tests. Three factors could have contributed:

1) English transliteration unit size ranges from 1 letter to 7 letters. The fixed size windows in ID3 obviously find difficult to capture the dynamics of various ranges. *n*-gram TM seems to have better captured the dynamics of transliteration units;

2) The backoff smoothing of *n*-gram TM is more effective than that of ID3;

3) Unlike *n*-gram TM, ID3 requires a separate aligning process for bilingual dictionary. The resulting alignment may not be optimal for tree construction. Nevertheless, ID3 presents another successful implementation of DOM framework.

5.2 DOM *vs.* phoneme-based approach

Due to lack of standard data sets, it is difficult to compare the performance of the *n*-gram TM to that of other approaches. For reference purpose, we list some reported studies on other databases of *E2C* transliteration tasks in Table 10. As in the references, only character and *Pinyin* error rates are reported, we only include our character and *Pinyin* error rates for easy reference. The reference data are extracted from Table 1 and 3 of (Virga and Khudanpur 2003). As we have not found any *C2E* result in the literature, only *E2C* results are compared here.

The first 4 setups by Virga *et al* all adopted the phoneme-based approach in the following steps:

1) English name to English phonemes;
2) English phonemes to Chinese *Pinyin*;
3) Chinese *Pinyin* to Chinese characters.

It is obvious that the *n*-gram TM compares favorably to other techniques. *n*-gram TM presents an error reduction of 74.6%=(42.5-10.8)/42.5% for *Pinyin* over the best reported result, Huge MT (Big MT) test case, which is noteworthy.

The DOM framework shows a quantum leap in performance with *n*-gram TM being the most successful implementation. The *n*-gram TM and ID3 under direct orthographic mapping (DOM) paradigm simplify the process and reduce the chances of conversion errors. As a result, *n*-gram TM and ID3 do not generate Chinese *Pinyin* as intermediate results. It is noted that in the 374 legitimate Chinese characters for transliteration, character to *Pinyin* mapping is unique while *Pinyin* to character mapping could be one to many. Since we have obtained results in character already, we expect less *Pinyin* error than character error should a character-to-*Pinyin* mapping be needed.

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%
ID3/DOM	34,777	2,896	< 15.6%	15.6%

Table 10. Performance reference in recent studies

6 Conclusions

In this paper, we propose a new framework (DOM) for transliteration. *n*-gram TM is a successful realization of DOM paradigm. It generates probabilistic orthographic transformation rules using a data driven approach. By skipping the intermediate phonemic interpretation, the transliteration error rate is reduced significantly.

Furthermore, 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. Unlike other related work where pre-alignment is needed, the new framework greatly reduces the development efforts of machine transliteration systems. Although the framework is implemented on an English-Chinese personal name data set, without loss of generality, it well applies to transliteration of other language pairs such as English/Korean and English/Japanese.

It is noted that place and company names are sometimes translated in combination of transliteration and meanings, for example, /Victoria-Fall/ becomes 维多利亚瀑布 (*Pinyin*:Wei Duo Li Ya Pu Bu). As the proposed framework allows direct orthographical mapping, it can also be easily extended to handle such name translation. We expect to see the proposed model to be further explored in other related areas.

References

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, pp138

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*, ASRU 2001

Jelinek, F. 1991, *Self-organized language modeling for speech recognition*, In Waibel, A. and Lee K.F. (eds), Readings in Speech Recognition, Morgan Kaufmann., San Mateo, CA

K. Knight and J. Graehl. 1998. *Machine Transliteration*, Computational Linguistics 24(4)

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

Quinlan J. R. 1993, *C4.5 Programs for machine learning*, Morgan Kaufmann , San Mateo, CA

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

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

The Onomastica Consortium, 1995. *The Onomastica interlanguage pronunciation lexicon*, Proceedings of EuroSpeech, Madrid, Spain, Vol. 1, pp829-832

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