# Machine Transliteration: Leveraging on Third Languages

**Min Zhang** **Xiangyu Duan** **Vladimir Pervouchine** **Haizhou Li**
Institute for Infocomm Research, A-STAR
{mzhang, xduan, vpervouchine, hli}@i2r.a-star.edu.sg

## Abstract

This paper presents two pivot strategies for statistical machine transliteration, namely *system-based* pivot strategy and *model-based* pivot strategy. Given two independent source-pivot and pivot-target name pair corpora, the *model-based* strategy learns a direct source-target transliteration model while the *system-based* strategy learns a source-pivot model and a pivot-target model, respectively. Experimental results on benchmark data show that the *system-based* pivot strategy is effective in reducing the high resource requirement of training corpus for low-density language pairs while the *model-based* pivot strategy performs worse than the *system-based* one.

## 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. This phonetic translation using computer is referred to as machine transliteration. With the rapid growth of the Internet data and the dramatic changes in the user demographics especially among the non-English speaking parts of the world, machine transliteration play a crucial role in most multilingual NLP, MT and CLIR applications (Hermjakob *et al.*, 2008; Mandl and Womser-Hacker, 2004). This is because proper names account for the majority of OOV issues and translation lexicons (even derived from large parallel corpora) usually fail to provide good coverage over diverse, dynamically increasing names across languages.

Much research effort has been done to address the transliteration issue in the research community (Knight and Graehl, 1998; Wan and Verspoor, 1998; Kang and Choi, 2000; Meng *et al.*, 2001; Al-Onaizan and Knight, 2002; Gao *et al.*, 2004; Klementiev and Roth, 2006; Sproat, 2006; Zelenko and Aone, 2006; Li *et al.*, 2004, 2009a, 2009b; Sherif and Kondrak, 2007; Bertoldi *et al.*, 2008; Goldwasser and Roth, 2008). These previous work can be categorized into three classes, i.e., grapheme-based, phoneme-based and hybrid methods. Grapheme-based method (Li *et al.*, 2004) treats transliteration as a direct orthographic mapping process and only uses orthography-related features while phoneme-based method (Knight and Graehl, 1998) treats transliteration as a phonetic mapping issue, converting source grapheme to source phoneme followed by a mapping from source phoneme to target phoneme/grapheme. Hybrid method in machine transliteration refers to the combination of several different models or decoders via re-ranking their outputs. The report of the first machine transliteration shared task (Li *et al.*, 2009a, 2009b) provides benchmarking data in diverse language pairs and systemically summarizes and compares different transliteration methods and systems using the benchmarking data.

Although promising results have been reported, one of major issues is that the state-of-the-art machine transliteration approaches rely heavily on significant source-target parallel name pair corpus to learn transliteration model. However, such corpora are not always availa-



*Coling 2010: Poster Volume*, pages 1444–1452,
Beijing, August 2010

ble and the amounts of the current available corpora, even for language pairs with English involved, are far from enough for training, letting alone many low-density language pairs. Indeed, transliteration corpora for most language pairs without English involved are unavailable and usually rather expensive to manually construct. However, to our knowledge, almost no previous work touches this issue.

To address the above issue, this paper presents two pivot language-based transliteration strategies for low-density language pairs. The first one is *system*-based strategy (Khapra *et al.*, 2010), which learns a source-pivot model from source-pivot data and a pivot-target model from pivot-target data, respectively. In decoding, it first transliterates a source name to *N*-best pivot names and then transliterates each pivot names to target names which are finally re-ranked using the combined two individual model scores. The second one is *model*-based strategy. It learns a direct source-target transliteration model from two independent[1] source-pivot and pivot-target name pair corpora, and then does direct source-target transliteration. We verify the proposed methods using the benchmarking data released by the NEWS2009[2] (Li *et al.*, 2009a, 2009b). Experiential results show that without relying on any source-target parallel data the system-based pivot strategy performs quite well while the model-based strategy is less effective in capturing the phonetic equivalent information.

The remainder of the paper is organized as follows. Section 2 introduces the baseline method. Section 3 discusses the two pivot language-based transliteration strategies. Experimental results are reported at section 4. Finally, we conclude the paper in section 5.

## 2 The Transliteration Model

Our study is targeted to be language-independent so that it can be applied to different language pairs without any adaptation effort. To achieve this goal, we use joint source-channel model (JSCM, also named as *n*-gram transliteration model) (Li *et la.*, 2004) under grapheme-based framework as our transliteration model due to its state-of-the-art performance by only using orthographical information (Li *et al.*, 2009a). In addition, unlike other feature-based methods, such as CRFs (Lafferty *et al.*, 2001), MaxEnt (Berger *et al.*, 1996) or SVM (Vapnik, 1995), the JSCM model directly computes model probabilities using maximum likelihood estimation (Dempster *et al.*, 1977). This property facilitates the implementation of the model-based strategy.

JSCM directly models how both source and target names can be generated simultaneously. Given a source name *S* and a target name *T*, it estimates the joint probability of *S* and *T* as follows:

$$\begin{aligned} P(S,T) &= P(s_1 \dots s_i \dots s_K, t_1 \dots t_i \dots t_K) \\ &= P(< s_1, t_1 >, \dots, < s_i, t_i >, \\ &\qquad \dots, < s_K, t_K >) \\ &= P(< s,t >_1, \dots, < s,t >_i, \\ &\qquad \dots < s,t >_K) \\ &= \prod_{k=1}^{K} P\left(< s,t >_k \mid < s,t >_1^{k-1}\right) \\ &\approx \prod_{k=1}^{K} P\left(< s,t >_k \mid < s,t >_{k-n+1}^{k-1}\right) \end{aligned}$$

where $s_i$ and $t_i$ is an aligned transliteration unit[3] pair, and *n* is the *n*-gram order.

In implementation, we compare different unsupervised transliteration alignment methods, including Giza++ (Och and Ney, 2003), the JSCM-based EM algorithm (Li *et al.*, 2004), the edit distance-based EM algorithm (Pervouchine *et al.*, 2009) and Oh et al.'s alignment tool (Oh *et al.*, 2009). Based on the aligned transliteration corpus, we simply learn the transliteration model using maximum likelihood estimation (Dempster *et al.*, 1977) and decode the transliteration result $T^* = \text{argmax}_T P(S,T)$ using stack decoder (Schwartz and Chow, 1990).

[1] Here "independent" means the source-pivot and pivot-target data are not derived from the same English name source.

[2] http://www.acl-ijcnlp-2009.org/workshops/NEWS2009/pages/sharedtask.html

[3] Transliteration unit is language dependent. It can be a Chinese character, a sub-string of English words, a Korean Hangual or a Japanese Kanji or several Japanese Katakanas.

## 3 Pivot Transliteration Strategies

### 3.1 System-based Strategy

The system-based strategy is first proposed by Khapra *et al.* (2010). They worked on system-based strategy together with CRF and did extensively empirical studies on Indic/Slavic/Semetic languages and English.

Given a source name *S*, a target name *T* and let *Z*(*S*, *Ź*) be the n-best transliterations of *S* in one or more pivot language *Ź* [4], the system-based transliteration strategy under JSCM can be formalized as follows:

$$\begin{aligned} P(S,T) &= \sum_{Ź} \sum_{Z(S,Ź)} P(S, Z(S,Ź), T) \\ &\approx \sum_{Z} P(S,Z,T) \\ &\approx \sum_{Z} P(T|S,Z) * P(S,Z) \end{aligned}$$

In the above formula, we assume that there is only one pivot language used in the derivation from the first line to the second line. Under the pivot transliteration framework, we can further simplify the above formula by assuming that $T$ is independent of $S$ when given $Z$. The assumption holds because the parallel name corpus between *S* and *T* is not available under the pivot transliteration framework. The n-best transliterations in pivot language are expected to be able to carry enough information of the source name *S* for translating *S* to target name *T*. Then, we have:

$$\begin{aligned} P(S,T) &= \sum_{Z} P(T|Z) * P(S,Z) \\ &= \sum_{Z} \frac{P(S,Z) * P(T,Z)}{P(Z)} \end{aligned} \quad (1)$$

Obviously we can train the two JSCMs of $P(S,Z)$ and $P(T,Z)$ using the two parallel corpora of $(S,Z)$ and $(T,Z)$, and train the language model $P(Z)$ using the monolingual corpus of $Z$. Following the nature of JSCM, Eq. (1) directly models how the source name *S* and pivot name *Z* and how the pivot name *Z* and the target name *T* are generated simultaneously. Since *Z* is considered twice in $P(S,Z)$ and $P(T,Z)$, the duplicated impact of *Z* is removed by dividing the model by $P(Z)$.

Given the model as described at Eq. (1), the decoder can be formulized as:

$$\begin{aligned} T^* &= \operatorname*{argmax}_T P(S,T) \\ &= \operatorname*{argmax}_T \left( \sum_Z \frac{P(S,Z) * P(T,Z)}{P(Z)} \right) \end{aligned} \quad (2)$$

If we consider multiple pivot languages, the modeling and decoding process are:

$$\begin{aligned} & P(S,T) \\ &= \sum_{Ź} \sum_{Z(S,Ź)} \left( \frac{P\left(S, Z(S,Ź)\right) * P(T, Z(S,Ź))}{P(Z(S,Ź))} \right) \end{aligned}$$

$$\begin{aligned} & T^* \\ &= \operatorname*{argmax}_T \left( \sum_{Ź} \sum_{Z(S,Ź)} \frac{P\left(S, Z(S,Ź)\right) * P\left(T, Z(S,Ź)\right)}{P\left(Z(S,Ź)\right)} \right) \end{aligned}$$

### 3.2 Model-based Strategy

Rather than combining the transitive transliteration results at system level, the model-based strategy aims to learn a direct model $P(S,T)$by combining the two individual models of $P(S,Z)$ and $P(T,Z)$, which are learned from the two parallel corpora of $(S,Z)$ and $(T,Z)$, respectively. Now let us use bigram as an example to illustrate how to learn the transliteration model $P(S,T) = \prod_{k=1}^{K} P(< s,t >_k \mid < s,t >_{k-1})$ using the model-based strategy.

$$\begin{aligned} & P(< s,t >_k \mid < s,t >_{k-1}) \\ &= \frac{P(< s,t >_k, < s,t >_{k-1})}{P(< s,t >_{k-1})} \end{aligned} \quad (3)$$

where,

$$\begin{aligned} & P(< s,t >_k, < s,t >_{k-1}) \\ &= P(s_k, s_{k-1}, t_k, t_{k-1}) \\ &= \sum_{z_k, z_{k-1}} P(s_k, s_{k-1}, t_k, t_{k-1}, z_k, z_{k-1}) \\ &= \sum_{z_k, z_{k-1}} P(t_k, t_{k-1} | s_k, s_{k-1}, z_k, z_{k-1}) \\ &\qquad * P(s_k, s_{k-1}, z_k, z_{k-1}) \end{aligned}$$

[4] There can be multiple pivot languages used in the two strategies. However, without loss of generality, we only use one pivot language to facilitate our discussion. It is very easy to extend one pivot language to multiple ones by considering all the pivot transliterations in all pivot languages.

The same as the system-based strategy, we can further simplify the above formula by assuming that $T$ is independent of $S$ when given $Z$. Indeed, $P(t_k, t_{k-1}|s_k, s_{k-1}, z_k, z_{k-1})$ cannot be estimated directly from training corpus. Then we have:

$$\begin{aligned} P(< s,t >_k, < s,t >_{k-1}) &= \sum_{z_k, z_{k-1}} P(t_k, t_{k-1}|s_k, s_{k-1}, z_k, z_{k-1}) \\ &\quad * P(s_k, s_{k-1}, z_k, z_{k-1}) \\ &\approx \sum_{z_k, z_{k-1}} P(t_k, t_{k-1}|z_k, z_{k-1}) \\ &\quad * P(s_k, s_{k-1}, z_k, z_{k-1}) \\ &\approx \sum_{z_k, z_{k-1}} P(t_k, t_{k-1}, z_k, z_{k-1}) \\ &\quad * P(s_k, s_{k-1}, z_k, z_{k-1}) \\ &\quad / P(z_k, z_{k-1}) \end{aligned} \quad (4)$$

where $P(t_i, t_{i-1}, z_i, z_{i-1})$, $P(s_i, s_{i-1}, z_i, z_{i-1})$ and $P(z_i, z_{i-1})$ can be directly learned from training corpus. $P(< s,t >_{k-1})$ for Eq (3) can also be estimatedas follows.

$$P(< s,t >_{k-1}) = \sum_{<s,t>_k} P(< s,t >_k, < s,t >_{k-1})$$

In summary, eq. (1) formulizes the system-based strategy and eq. (3), (4) and (5) formulize the model-based strategy, where we can find that they share the same nature of generating source, pivot and target names simultaneously. The difference is that the model-based strategy operates at fine-grained transliteration unit level.

### 3.3 Comparison with Previous Work

Almost all previous work on machine transliteration focuses on direct transliteration or transliteration system combination. There is only one recent work (Khapra *et al.*, 2010) touching this issue. They work on system-based strategy together with CRF. Compared with their work, this paper gives more formal definitions and derivations of system-based strategy from modeling and decoding viewpoints based on the JSCM model.

The pivot-based strategies at both system and model levels have been explored in machine translation. Bertoldi *et al.* (2008) studies two pivot approaches for phrase-based statistical machine translation. One is at system level and one is to re-construct source-target data and alignments through pivot data. Cohn and Lapata (2007) explores how to utilize multilingual parallel data (rather than pivot data) to improve translation performance. Wu and Wang (2007, 2009) extensively studies the model-level pivot approach and also explores how to leverage on rule-based translation results in pivot language to improve translation performance. Utiyama and Isahara (2007) compares different pivot approaches for phrase-based statistical machine translation. All of the previous work on machine translation works on phrase-based statistical machine translation. Therefore, their translation model is to calculate phrase-based conditional probabilities at unigram level ($P(t_k|s_k)$) while our transliteration model is to calculate joint transliteration unit-based conditional probabilities at bigram level ($P(< s,t >_k | < s,t >_{k-1})$).

## 4 Experimental Results

### 4.1 Experimental Settings

We use the NEWS 2009 benchmark data as our experimental data (Li *et al.*, 2009). The NEWS 2009 data includes 8 language pairs, where we select English to Chinese/Japanese /Korean data (E-C/J/K) and based on which we further construct Chinese to Japanese/Korean and Japanese to Korean for our data.

| Language Pair | Training | Dev | Test |
|---|---|---|---|
| English-Chinese | 31,961 | 2896 | 2896 |
| English-Japanese | 23,225 | 1492 | 1489 |
| English-Korean | 4,785 | 987 | 989 |
| Chinese-Japanese | 12,417 | 75 | 77 |
| Chinese-Korean | 2,148 | 32 | 31 |
| Japanese-Korean | 6,035 | 65 | 69 |

Table 1. Statistics on the data set

Table 1 reports the statistics of all the experimental data. To have a more accurate evaluation, the test sets have been cleaned up to make sure that there is no overlapping between any test set with any training set. In addition, the three E-C/J/K data are generated independently so that there is very small percentage of over-

lapping between them. This can ensures the evaluation of the pivot study fair and accurate.

We compare different alignment algorithms on the DEV set. Finally we use Pervouchine *et al.* (2009)'s alignment algorithm for Chinese-English/Japanese/Korean and Oh *et al.* (2009)'s alignment algorithm for English-Korean and Li *et al.* (2004)'s alignment algorithm for English-Japanese and Japanese-Korean. Given the aligned corpora, we directly learn each individual JSCM model (i.e., *n*-gram transliteration model) using SRILM toolkits (Stolcke, 2002). We also use SRILM toolkits to do decoding. For the system-based strategy, we output top-20 pivot transliteration results.

For the evaluation matrix, we mainly use top-1 accuracy (ACC) (Li *et al.*, 2009a) to measure transliteration performance. For reference purpose, we also report the performance using all the other evaluation matrixes used in NEWS 2009 benchmarking (Li *et al.*, 2009a), including F-score, MRR, MAP_ref, MAP_10 and MAP_sys. It is reported that F-score has less correlation with other matrixes (Li *et al.*, 2009a).

## 4.2 Experimental Results

### 4.2.1 Results of Direct Transliteration

Table 2 reports the performance of direct transliteration. The first three experiments (line 1-3) are part of the NEWS 2009 share tasks and the others are our additional experiments for our pivot studies.

Comparison of the first three experimental results and the results reported at NEWS 2009 shows that we achieve comparable performance with their best-reported systems at the same conditions of using single system and orthographic features only. This indicates that our baseline represents the state-of-the-art performance. In addition, we find that the *back*-transliteration (line 4-6) consistently performs worse than its corresponding *forward*-transliteration (line 1-3). This observation is consistent with what reported at previous work (Li *et al.*, 2004; Zhang *et al.*, 2004). The main reason is because English has much more transliteration units than foreign C/J/K languages. This makes the transliteration from English to C/J/K a many-to-few mapping issue and *back*-transliteration a few-to-many mapping issue. Therefore *back*-transliteration has more ambiguities and thus is more difficult.

Overall, the lower six experiments (line 7-12) shows worse performance than the upper six experiments which has English involved. This is mainly due to the less available training data for the language pairs without English involved. This observation motivates our study using pivot language for machine transliteration.

### 4.2.2 Results of System-based Strategy

Table 3 reports three empirical studies of system-based strategies: Japanese to Chinese through English, Chinese to Japanese through English and Chinese to Korean through English. Considering the fact that those language pairs with English involved have the most training data, we select English as pivot language in the system-based study. Table 3 clearly shows that:

- The system-based pivot strategy is very effective, achieving significant performance improvement over the direct transliteration by 0.09, 0.07 and 0.03 point of ACC in the three language pairs, respectively;
- Different from other pipeline methodologies, the system-based pivot strategy does not suffer heavily from the error propagation issue. Its ACC is significantly better than the product of the ACCs of the two individual systems;
- The combination of pivot system and direct system slightly improves overall ACC.

We then conduct more experiments to figure out the reasons. Our further statistics and analysis show the following reasons for the above observations:

The pivot approach is able to use source-pivot and pivot-target data whose amount is much more than that of the available direct source-target data.

- The nature of transliteration is phonetic translation. Therefore a little bit variation in orthography may not hurt or even help to improve transliteration performance in some cases as long as the orthographical variations keep the phonetic equivalent

| Language Pairs | ACC | F-Score | MRR | MAP_ref | MAP_10 | MAP_sys |
|---|---|---|---|---|---|---|
| English → Chinese | 0.678867 | 0.871497 | 0.771563 | 0.678867 | 0.252382 | 0.252382 |
| English → Japanese | 0.482203 | 0.831983 | 0.594235 | 0.471766 | 0.201510 | 0.201510 |
| English → Korean | 0.439838 | 0.722365 | 0.543039 | 0.439585 | 0.171621 | 0.171621 |
| Chinese → English | 0.395250 | 0.867702 | 0.518292 | 0.372403 | 0.222787 | 0.222787 |
| Japanese → English | 0.334839 | 0.838212 | 0.450984 | 0.319277 | 0.168032 | 0.168032 |
| Korean → English | 0.088505 | 0.494205 | 0.109249 | 0.088759 | 0.034380 | 0.034380 |
| Chinese → Japanese | 0.385965 | 0.769245 | 0.473851 | 0.348319 | 0.159948 | 0.159948 |
| Japanese → Chinese | 0.402597 | 0.714193 | 0.491595 | 0.402597 | 0.165581 | 0.165581 |
| Chinese → Korean | 0.290323 | 0.571587 | 0.341129 | 0.290323 | 0.178652 | 0.178652 |
| Korean → Chinese | 0.129032 | 0.280645 | 0.156042 | 0.129032 | 0.048163 | 0.048163 |
| Japanese → Korean | 0.313433 | 0.678240 | 0.422862 | 0.313433 | 0.208310 | 0.208310 |
| Korean → Japanese | 0.089286 | 0.321617 | 0.143948 | 0.091270 | 0.049992 | 0.049992 |

Table 2. Performance of direct transliterations

| Language Pairs | ACC | F-Score | MRR | MAP_ref | MAP_10 | MAP_sys |
|---|---|---|---|---|---|---|
| Jap→Eng→Chi (**Pivot**) | 0.493506 | 0.750711 | 0.617440 | 0.493506 | 0.195151 | 0.195151 |
| Jap→Eng→Chi (**Pivot**) + Jap → Chi (**Direct**) | 0.506494 | 0.753958 | 0.622851 | 0.506494 | 0.196017 | 0.196017 |
| Jap → Chi (**Direct**) | 0.402597 | 0.714193 | 0.491595 | 0.402597 | 0.165581 | 0.165581 |
| Jap → Eng (**Direct**) | 0.334839 | 0.838212 | 0.450984 | 0.319277 | 0.168032 | 0.168032 |
| Eng → Chi (**Direct**) | 0.678867 | 0.871497 | 0.771563 | 0.678867 | 0.252382 | 0.252382 |
| Chi→Eng→Jap (**Pivot**) | 0.456140 | 0.777494 | 0.536591 | 0.414961 | 0.183222 | 0.183222 |
| Chi→Eng→Jap (**Pivot**) + Chi → Jap (**Direct**) | 0.491228 | 0.801443 | 0.563297 | 0.450049 | 0.191742 | 0.191742 |
| Chi → Jap (**Direct**) | 0.385965 | 0.769245 | 0.473851 | 0.348319 | 0.159948 | 0.159948 |
| Chi → Eng (**Direct**) | 0.395250 | 0.867702 | 0.518292 | 0.372403 | 0.222787 | 0.222787 |
| Eng → Jap (**Direct**) | 0.482203 | 0.831983 | 0.594235 | 0.471766 | 0.201510 | 0.201510 |
| Chi→Eng→Kor (**Pivot**) | 0.322581 | 0.628146 | 0.432642 | 0.322581 | 0.175822 | 0.175822 |
| Chi→Eng→Kor (**Pivot**) + Chi → Kor (**Direct**) | 0.331631 | 0.632967 | 0.439143 | 0.334222 | 0.176543 | 0.176543 |
| Chi → Kor (**Direct**) | 0.290323 | 0.571587 | 0.341129 | 0.290323 | 0.178652 | 0.178652 |
| Chi → Eng (**Direct**) | 0.395250 | 0.867702 | 0.518292 | 0.372403 | 0.222787 | 0.222787 |
| Eng → Kor (**Direct**) | 0.439838 | 0.722365 | 0.543039 | 0.439585 | 0.171621 | 0.171621 |

Table 3. Performance comparison of system-based strategy on Jap (Japanese) to Chi (Chinese) and Chi (Chinese) to Jap (Japanese)/Kor (Korean) through Eng (English) as pivot language, where "…(**Pivot**) + …(**Direct**)" means that for the same language pair we merge and re-rank the pivot transliteration and direct transliteration results

information. Indeed, given one source English names, there are usually more than one correct transliteration references in Japanese/Korean. This case also happens to English to Chinese although not so heavy as in English to Japanese/Korean.

| Language Pairs | ACC | F-Score | MRR | MAP_ref | MAP_10 | MAP_sys |
|---|---|---|---|---|---|---|
| Chi→Eng→Jap (**Model-based Pivot: O**) | 0.087719 | 0.538454 | 0.117446 | 0.085770 | 0.040645 | 0.040645 |
| Chi→Eng→Jap (**Model-based Pivot: R**) | 0.210526 | 0.746497 | 0.381210 | 0.201267 | 0.156106 | 0.156106 |
| Chi→Eng→Jap (**System-based Pivot**) | 0.456140 | 0.777494 | 0.536591 | 0.414961 | 0.183222 | 0.183222 |
| Chi → Jap (**Direct**) | 0.385965 | 0.769245 | 0.473851 | 0.348319 | 0.159948 | 0.159948 |
| Jap→Chi→Eng (**Model-based Pivot**) | 0.148504 | 0.724623 | 0.224253 | 0.141791 | 0.088966 | 0.088966 |
| Jap→Chi→Eng (**System-based Pivot**) | 0.201581 | 0.741627 | 0.266507 | 0.191926 | 0.098024 | 0.134730 |
| Jap → Eng (**Direct**) | 0.334839 | 0.838212 | 0.450984 | 0.319277 | 0.168032 | 0.168032 |
| Eng→Jap→Kor (**Model-based Pivot**) | 0.206269 | 0.547732 | 0.300641 | 0.206269 | 0.145882 | 0.145882 |
| Eng→Jap→Kor (**System-based Pivot**) | 0.315470 | 0.629640 | 0.404769 | 0.315723 | 0.167587 | 0.225892 |
| Eng → Kor (**Direct**) | 0.439838 | 0.722365 | 0.543039 | 0.439585 | 0.171621 | 0.171621 |

Table 4. Performance of Model-based Pivot Transliteration Strategy

- The N-best accuracy of machine transliteration (of both to and from English) is very high[5]. It means that in most cases the correct transliteration in pivot language can be found in the top-20 results and the other 19 results hold the similar pronunciations with the correct one, which can serve as alternative "quasi-correct" inputs to the second stage transliterations and thus largely improve the overall accuracy.

The above analysis holds when using English as pivot language. Now let us see the case of using non-English as pivot language. Table 4 reports two system-based strategies using Chinese and Japanese as pivot languages, where we can find that the performance of two system-based strategies is worse than that of the direct transliterations. The main reason is because that the direct transliteration utilizes much more training data than the pivot approach. However, the good thing is that the system-based pivot strategy using non-English as pivot language still does not suffer from error propagation issue. Its ACC is significantly better than the product of the ACCs of the two individual systems.

#### 4.2.3 Results of Model-based Strategy

Table 4 reports the performance of model-based strategy. It clearly shows that the model-based strategy is less effective and performs much worse than both the system-based strategy and direct transliteration.

While the model-based strategy works well at phrase-based statistical machine translation (Wu and Wang, 2007, 2009), it does not work at machine transliteration. To investigate the reasons, we conduct many additional experiments and do statistics on the model and

[5] Both our studies and previous work (Li et al., 2004; Zhang et al., 2004) shows that the top-20 accuracy from English to J/K is more than 0.85 and more than 0.95 in English-Chinese case. The top-20 accuracy is a little worse from C/J/K to English, but still more than 0.7.

aligned training data[6]. From this in-depth analysis, we find that main reason is due to the fact that the model-based strategy introduces too many entries (ambiguities) to the final transliteration model. For example, in the Jap→Chi→Eng experiment, the unigram and bigram entries of the transliteration model obtained by the model-based strategy are 45 and 6.6 times larger than that of the transliteration model trained directly from parallel data. This is not surprising. Given a transliteration unit in pivot language, it can generate $m * n$ source-to-target transliteration unit mappings (unigram entry of the model), where $m$ is the number of the source units that can be mapped to the pivot unit and $n$ is the number of the target units that can be mapped from the pivot unit.

Besides the ambiguities introduced by the large amount of entries in the model, another reason that leads to the worse performance of model-based strategy is the size inconsistence of transliteration unit of pivot language. As shown at Table 4, we conduct three experiments. In the first experiment (Chi→Eng→Jap), we use English as pivot language. We find that the English transliteration unit size in Chi→Eng model is much larger than that in Eng→Jap model. This is because from phonetic viewpoint, in Chi→Eng model, the English unit is at syllable level (corresponding one Chinese character) while in Eng→Jap model, the English unit is at sub-syllable level (consonant or vowel or syllable, corresponding one Japanese Katakana). This is the reason why we conduct two model-based experiments for Chi→Eng→Jap. One is based on the original alignments (**Model-based Pivot: O**) and one is based on the reconstructed alignments[7] (**Model-based Pivot: R**). Experimental results clearly show that the reconstruction improves performance significantly. In the second and third experiments (Jap→Chi→Eng, Eng→Jap→Kor), we use Chinese and Japanese as pivot languages. Therefore we do not need to re-construct transliteration units and alignments. However, the performance is still very poor. This is due to the first reason of the large amount of ambiguities.

The above two reasons (ambiguities and transliteration unit inconsistence) are mixed together, leading to the worse performance of the model-based strategy. We believe that the fundamental reason is because the pivot transliteration unit is too small to be able to convey enough phonetic information of source language to target language and thus generates too many alignments and ambiguities.

## 5 Conclusions

A big challenge to statistical-based machine transliteration is the lack of the training data, esp. to those language pairs without English involved. To address this issue, inspired by the research in the SMT research community, we study two pivot transliteration methods. One is at system level while another one is at model level. We conduct extensive experiments using NEW 2009 benchmarking data. Experimental results show that system-based method is very effective in capturing the phonetic information of source language. It not only avoids successfully the error propagation issue, but also further boosts the transliteration performance by generating more alternative pivot results as the inputs of the second stage. In contrast, the model-based method in its current form fails to convey enough phonetic information from source language to target language.

For the future work, we plan to study how to improve the model-based strategy by pruning out the so-called "bad" transliteration unit pairs and re-sampling the so-called "good" unit pairs for better model parameters. In addition, we also would like to explore other pivot-based transliteration methods, such as constructing source-target training data through pivot languages.

## References

Yaser Al-Onaizan and Kevin Knight. 2002. *Translating named entities using monolingual and bilingual resources*. ACL-02

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

[6] However, due to space limitation, we are not allowed to report the details of those experiments.

[7] Based on the English transliteration units obtained from Chi →Eng, we reconstruct the English transliteration units and alignments in Eng→Jap by merging the adjacent units of both English and Japanese to syllable level.

N. Bertoldi, M. Barbaiant, M. Federico and R. Cattoni. 2008. *Phrase-based Statistical Machine Translation with Pivot Languages*. IWSLT-08

Trevor Cohn and Mirella Lapata. 2007. *Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora*. ACL-07

Andrew Finch and Eiichiro Sumita. 2008. *Phrase-based machine transliteration*. IJCNLP-08

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

Dan Goldwasser and Dan Roth. 2008. *Transliteration as constrained optimization*. EMNLP-08

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

Ulf Hermjakob, K. Knight and Hal Daum ́e. 2008. *Name translation in statistical machine translation: Learning when to transliterate*. ACL-08

John Lafferty, Fernando Pereira, Andrew McCallum. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. ICML-01

B.J. Kang and Key-Sun Choi. 2000. *Automatic Transliteration and Back-transliteration by Decision Tree Learning*. LREC-00

Mitesh Khapra, Kumaran A and Pushpak Bhattacharyya. 2010. *Everybody loves a rich cousin: An empirical study of transliteration through bridge languages.* NAACL-HLT-10

Alexandre Klementiev and Dan Roth. 2006. *Weakly supervised named entity transliteration and discovery from multilingual comparable corpora*. COLING-ACL-06

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

P. Koehn, F. J. Och and D. Marcu. 2003. *Statistical phrase-based translation*. HLT-NAACL-03

J. Lafferty, A. McCallum and F. Pereira. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. ICML-01

Haizhou Li, A Kumaran, Vladimir Pervouchine and Min Zhang. 2009a. *Report of NEWS 2009 Machine Transliteration Shared Task*. IJCNLP-ACL-09 Workshop: NEWS-09

Haizhou Li, A Kumaran, Min Zhang and Vladimir Pervouchine. 2009b. *Whitepaper of NEWS 2009 Machine Transliteration Shared Task*. IJCNLP-ACL-09 Workshop: NEWS-09

Haizhou Li, Ming Zhang and Jian Su. 2004. *A Joint Source-Channel Model for Machine Transliteration*. ACL-04

Thomas Mandl and Christa Womser-Hacker. 2004. *How do Named Entities Contribute to Retrieval Effectiveness?* CLEF-04

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-01

Jong-Hoon Oh, Kiyotaka Uchimoto, and k. Torisawa. 2009. *Machine Transliteration with Target-Language Grapheme and Phoneme: Multi-Engine Transliteration Approach.* NEWS 2009

Franz Josef Och and Hermann Ney. 2003. *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics 29(1)

V. Pervouchine, H. Li and B. Lin. 2009. *Transliteration Alignment*. ACL-IJCNLP-09

R. Schwartz and Y. L. Chow. 1990. *The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypothesis*, ICASSP-90

Tarek Sherif and Grzegorz Kondrak. 2007. *Substring-based transliteration*. ACL-07

Richard Sproat, Tao Tao and ChengXiang Zhai. 2006. *Named entity transliteration with comparable corpora*. COLING-ACL-06

Andreas Stolcke. 2002. *SRILM - an extensible language modeling toolkit.* ICSLP-02

Masao Utiyama and Hitoshi Isahara. 2007. *A Comparison of Pivot Methods for Phrase-based Statistical Machine Translation*. NAACL-HLT-07

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer

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

Hua Wu and Haifeng Wang. 2007. *Pivot Language Approach for Phrase-based Statistical Machine Translation*. ACL-07

Hua Wu and Haifeng Wang. 2009. *Revisiting Pivot Language Approach for Machine Translation*. ACL-09

Dmitry Zelenko and Chinatsu Aone. 2006. *Discriminative methods for transliteration*. EMNLP-06

Min Zhang, Haizhou Li and Jian Su. 2004. *Direct Orthographical Mapping for machine transliteration*. COLING-04