# Joint Alignment and Artificial Data Generation: An Empirical Study of Pivot-based Machine Transliteration

**Min Zhang[1]    Xiangyu Duan[1]    Ming Liu[1]    Yunqing Xia[2]    Haizhou Li[1]**

[1]Institute for Infocomm Research, A-STAR, Singapore
`{mzhang,xduan,mliu,hli}@i2r.a-star.edu.sg`

[2]Dept. of Comp. Sci. & Tech., Tsinghua University, Beijing
`yqxia@tsinghua.edu.cn`

## Abstract

In this paper, we first carry out an investigation on two existing pivot strategies for statistical machine transliteration, namely *system*-based and *model*-based strategies, to figure out the reason why the previous model-based strategy performs much worse than the system-based one. We then propose a joint alignment algorithm to optimize transliteration alignments jointly across source, pivot and target languages to improve the performance of the model-based strategy. In addition, we further propose a novel *synthetic data*-based strategy, which artificially generates source-target data using pivot language. Experimental results on benchmarking data show that the proposed joint alignment optimization algorithm significantly improves the accuracy of model-based strategy and the proposed synthetic data-based strategy is very effective for pivot-based machine transliteration.

## 1   Introduction

Machine transliteration refers to the phonetic translation of names across languages by computer. 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 is important in many cross-lingual NLP, MT and CLIR applications as their performances have been shown to positively correlate with the correct conversion of names between the languages in several studies (Demner-Fushman and Oard, 2002; Mandl and Womser-Hacker, 2005; Hermjakob *et al.*, 2008; Udupa *et al.*, 2009). However, the traditional source for name equivalence, the bilingual dictionaries — whether handcrafted or statistical built — offer only limited support because new names always emerge.

All of the above points to the critical need for high-performance machine transliteration technology. Much research effort has been made to address this issue in the research community (Knight and Graehl, 1998; Meng *et al.*, 2001; Al-Onaizan and Knight, 2002; Oh and Choi, 2002; 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 falls into three categories, i.e., grapheme-based, phoneme-based and hybrid methods (Li *et al.*, 2009a, 2009b). The report of the first machine transliteration shared task NEWS 2009 (Li *et al.*, 2009a, 2009b) provides common benchmarking data in diverse language pairs and systematically evaluate the state-of-the-art technologies using their provided data.

Although promising results have been reported, one of major issues is that the current transliteration methods rely heavily on significant amount of source-target parallel data to learn transliteration model. However, such corpora are not always available and the amounts of the currently 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 are usually rather expensive to manually construct (Khapra *et al.*, 2010; Zhang *et al.*, 2010). To date, only two previous works (Khapra *et al.*, 2010; Zhang *et al.*, 2010) touch this issue of transliterating names across low-density language pairs. Both of them resort to pivot language-based approaches to address this issue.

Khapra *et al.* (2010) proposes the system-based pivot strategy for machine transliteration, 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 name to target names which are finally re-ranked using the combined two individual transliteration scores. Zhang *et al.* (2010) verifies the system-based strategy together with joint source-channel model (Li *et al.*, 2004) on Chinese, English, Korean and Japanese data (Li *et al.*, 2009a, 2009b) and they further proposes a model-based strategy, which learns a direct source-target transliteration model from two independent [1] source-pivot and pivot-target name pair corpora, and does direct source-target decoding without relying on pivot languages. However, it was reported that the model-based strategy performed much worse than the system-based one (Zhang *et al.*, 2010).

This paper investigates the reason why previous model-based strategy performs worse than system-based one and then proposes a joint alignment algorithm to solve the alignment unit inconsistent issue, which is the main reason of leading to the worse performance of model-based strategy. The key point of the proposed joint alignment algorithm is to jointly optimize transliteration unit alignments among source, pivot and target languages. In addition, the paper further proposes a novel synthetic data-based strategy for pivot-based machine transliteration. It automatically constructs source-target data using source-pivot and pivot-target data, and then trains a direct source-target transliteration model using the synthetic data. We verify the proposed methods using the benchmarking data released at NEWS2009 (Li *et al.*, 2009a, 2009b). Experiential results show that our proposed joint alignment optimization algorithm is able to effectively solve the transliteration unit mismatching issue and the proposed synthetic data-based strategy is very effective, achieving the best-reported performance.

The rest of the paper is organized as follows. Section 2 introduces the direct transliteration model. Section 3 discusses our proposed joint alignment algorithm and synthetic data-based strategy. Experimental results are reported at section 4. Finally, we conclude the paper in section 5.

## 2 The Transliteration Model: JSCM

To make our study language-independent, we select joint source-channel model (JSCM, also named as *n*-gram transliteration model) (Li *et al.*, 2004) under grapheme-based framework as our transliteration model due to its state-of-the-art performance and using orthographical information only (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:

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

where $s_i$ and $t_i$ is an aligned transliteration unit[2] pair, and *n* is the *n*-gram order.

In our implementation, we compare different unsupervised transliteration alignment methods, including Giza++ (Och and Ney, 2003), JSCM-based EM algorithm (Li *et al.*, 2004), 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 learn the transliteration model using maximum likelihood estimation (Dempster *et al.*, 1977) and decode the transliteration result $T^* = \operatorname{argmax}_T P(S,T)$ using stack decoder (Schwartz and Chow, 1990).

---

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

[2] 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 Joint Alignment and Synthetic Data-based Strategy

In this section, we elaborate our proposed joint alignment algorithm and synthetic data-based strategy for pivot-based machine transliteration.

### 3.1 System-based Strategy

Given a source name $S$, a target name $T$ and let $Z$ be the n-best transliterations of $S$ in a pivot language $\acute{Z}$ [3], the system-based transliteration strategy under JSCM can be formulized as follows:

$$
\begin{aligned}
P(S,T) &= \sum_Z P(S,Z,T) \\
&= \sum_Z P(T|S,Z) * P(S,Z) \\
&\approx \sum_Z P(T|Z) * P(S,Z) \\
&\approx \sum_Z \frac{P(S,Z) * P(T,Z)}{P(Z)}
\end{aligned}
\tag{2}
$$

where $P(S,Z)$ and $P(T,Z)$ can be computed using JSCM as formalized at Eq. (1). Eq. (2) assumes that $T$ is independent of $S$ when given $Z$ 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 $S$. Following the nature of JSCM, Eq. (2) 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 being divided by $P(Z)$.

### 3.2 Joint Alignment Algorithm for 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 $P(T,Z)$. Here we use bigram as an example to illustrate how to learn the JSCM transliteration model $P(S,T) = \prod_{k=1}^K P(<s,t>_k \mid <s,t>_{k-1})$ using the model-based strategy.

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

---
[3] There can be multiple pivot languages used. However, same as Khapra *et al.* (2010) and Zhang *et al.* (2010), without loss of generality, we only use one pivot language to facilitate our discussion. It is straightforward to extend one pivot language to multiple ones by considering all the pivot transliterations in all pivot languages.

$$
= \frac{P(<s,t>_k, <s,t>_{k-1})}{P(<s,t>_{k-1})}
\tag{3}
$$

where,

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

$$
P(<s,t>_{k-1}) = \sum_{<s,t>_k} P(<s,t>_k, <s,t>_{k-1})
\tag{5}
$$

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 estimated at training corpus.

In summary, eq. (2) formalizes the system-based strategy while eq. (3) formalizes the model-based strategy, where we can find that eq. (2) involves the pivot language $Z$ in modeling and decoding while eq. (3) does not (its model parameters are pre-computed using eq. (4) and (5) during training).

In previous work (Zhang *et al.*, 2010), the model-based strategy was reported to perform much worse than the system-based. We find that the main reason is due to the size inconsistence of transliteration unit of pivot language in the source-pivot and pivot-target alignments during training. As shown at eq. (4), the source-target model is calculated using the source-pivot el$P(s_k, s_{k-1}, z_k, z_{k-1})$and the pivot-target model $P(t_k, t_{k-1}, z_k, z_{k-1})$ directly. This requests that the pivot transliteration unit $z_k, z_{k-1}$ must be consistent in the two individual modes. Thus, all the source-pivot and pivot-target model parameters $P(*_k, *_{k-1}, z_k, z_{k-1})$ are of no use if their involved pivot unit $z_k, z_{k-1}$ can only be found at either source-pivot or pivot-target model. Unfortunately in the only previous work (Zhang *et al.*, 2010), the source-pivot model and pivot-target

model are trained separately, i.e., their object function is to maximize $P(S, Z)$ and $P(T, Z)$ independently. This results in serious pivot transliteration unit inconsistent issue for some language pairs. For example, in our experiment (Chinese→English→ Japanese) with English as pivot language, we find that the English transliteration unit size in Chinese→English model is much larger than that in English→Japanese model. This is because from phonetic viewpoint, in Chinese→English model, the English unit is at syllable level (corresponding one Chinese character) while in English→Japanese model, the English unit is at sub-syllable level (consonant or vowel or syllable, corresponding one Japanese Katakana). Following example excerpted from our training corpus illustrates the pivot transliteration unit mismatching issue, where the English word "Aachen" is segmented into "Aa" and "chen" in Chinese-English model while it is segmented into "A", "a", "che" and "n" in English-Japanese model. This trilingual pair is then of no use in model-based strategy.

To solve the mismatching issue, this paper proposes a joint alignment algorithm to jointly optimize transliteration unit alignments among source, pivot and target languages for model-based strategy. To facilitate discussion, we base on the task of using English as pivot language for Chinese-Japanese transliteration (see Table 1) to present our proposed algorithm. The core idea of this algorithm is to use Chinese-English alignments as a constraint to do English-Japanese alignment. The algorithm consists of the following 6 steps:

**Algorithm 1.  Joint Alignment**

**Inputs:**
    Chinese-English Name List (CE).
    English-Japanese Name List (EJ).

**Outputs:**
    More consistent CE and EJ alignments at Chinese syllable level and a direct Chinese-Japanese (CJ) JSCM.

1. Align the CE names at Chinese syllable level using the JSCM-based EM algorithm (Li *et al.*, 2004).
2. Train a transliteration unit-based English bi-gram LM with the transliteration unit-segmented (at step 1) English side names of CE using SRILM toolkits (Stolcke, 2002). Note that here the English transliteration units are corresponding to Chinese syllable level.

3. Align the Chinese-English-Japanese (CEJ) names that are the intersection of the entire CE and EJ names (with the same English names).
   a. CE part of CEJ has been aligned at Step 1.
   b. Align the CJ part of CEJ at Chinese syllable level using the JSCM-based EM algorithm (Li *et al.*, 2004).
   c. Construct the CEJ name alignments by

**Chinese:**  亚(*ya*) 琛(*chen*)

**English:**  A a c h e n

**Japanese:**  ア 一 ヘ ン
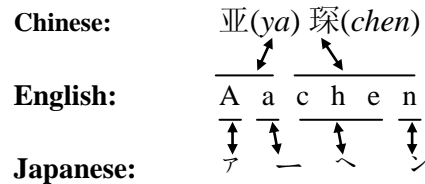
   merging the CE and CJ alignments.
4. Align EJ names at Chinese syllable level.
   a. The intersection part of EJ with CJ (*i*EJ) has already been aligned at Chinese syllable level at step 3.
   b. Align the remaining non-intersection part of EJ name pairs (*ni*EJ) using Algorithm 2 with the help of the aligned intersection part done at step 4.a and the transliteration unit-based English bi-gram LM learned at step 2.
   c. Merge the above two parts.
5. Train two individual JSCMs using the Chinese syllable level-aligned CE and EJ name corpus, respectively.
6. Train a direct CJ JSCM using the two individual JSCMs learned at step 5 by the model-based pivot strategy as formulated at eqs. (3), (4) and (5).

**Algorithm 2.  Constrained EM-based Alignment**

**Inputs:**
   1. Non-intersection part of EJ name pairs (*ni*EJ).
   2. Intersection part of EJ name pairs (*i*EJ) aligned at Chinese syllable level and the initial JSCM (named as iJSCM) learned from this corpus (step 3.d of Algorithm 1).
   3. The transliteration unit-based English bi-gram LM (named as eLM, step 2 of Algorithm 1).

**Output:**
   English-Japanese name pairs aligned at Chinese syllable level.

1. Bootstrap initial alignment of the *ni*EJ name using the initial model iJSCM.

2. **Expectation**: re-train iJSCM using both the input *i*EJ name alignments and the updated *ni*EJ name alignments.
3. **Maximization**: Apply the re-trained iJSCM and the input eLM to obtain new alignments of the *ni*EJ names. Note that different from previous EM-based transliteration alignment algorithm (Li *et al.*, 2004) that only maximizes the JSCM probabilities, here we maximize two kinds of probabilities:

$$A^* = \underset{A}{\mathrm{argmax}}\, P(E, J, A) * P(E, A) \qquad (6)$$

where $E$ refers to English and $J$ refers to Japanese; $A$ is an alignment which defining the transliteration unit segmentations of $E$ and $J$, and their mappings; $P(E, J, A)$ is the JSCM probability of $E$ and $J$ under $A$; and $P(E, A)$ is the eLM transliteration unit bigram probability of $E$ segmented by $A$.
4. Go to step 2 until the alignment converges.
5. Output the *ni*EJ name alignments.

The motivation of the joint alignment algorithm (Algorithm 1 and 2) is to address the English transliteration unit mismatching issue by aligning the EJ at Chinese syllable level with the help of CE alignment. While the mismatching issue in the intersection part of the data is easy to solve by step 3 of Algorithm 1, it is more complicated at the non-intersection part. As illustrated at step 3 of Algorithm 2, the core idea is to use English segmentation learned from CE alignment (step 2 of Algorithm 1) and already-aligned intersection part of EJ (*i*EJ, step 3 of Algorithm 1) to constrain the EM alignment process. Therefore, the English bigram LM and the aligned intersection part (*i*EJ) keep unchanged during all the EM iterations. But in E step (step 2 of Algorithm 2), the iJSCM model is updated at each iteration using the entire EJ data while in M step (step 3 of Algorithm 2), the alignments are decoded out using both the iJSCM and the English bi-gram LM. Indeed, in our implementation, we introduce more knowledge sources, including transliteration unit insertion penalty and Japanese LM, into the M step by simply considering these two features at eq. (6).

Given the jointly optimized CE and EJ aligned name corpus, we can easily learn a direct CJ model using the pivot-based strategy (steps 5 and 6 of Algorithm 1).

## 3.3 Synthetic Data-based Strategy

Different from previous two strategies, the synthetic data-based strategy automatically constructs source-target data using source-pivot and pivot-target data, and then trains a direct source-target transliteration model using the synthetic and any other available source-target data. The philosophy of this strategy is straightforward while the key is how to generate "good" data. Next, we also use Chinese-English-Japanese as example to elaborate this strategy.

## Algorithm 3. Artificial Data Generation

**Inputs:**
  Chinese-English Name List (CE).
  English-Japanese Name List (EJ).

**Outputs:**
  Synthetic Chinese-Japanese name pairs.

1. Directly output those CJ names (*i*CJ), which are the intersection of the entire CE and EJ names (with the same English names).
2. Transliterate those Chinese names which are not in *i*CJ to Japanese using either the system-based or model-based strategy. To maintain the transliteration quality, we consider both forward and backward transliteration probabilities as well as the information whether the original Chinese can be recovered from a transliterated Japanese name. The process is formalized as follows:

$$J^* = \underset{J}{\mathrm{argmax}}\, P(J|C) * P(\acute{C}|J) * I(\acute{C}, C) * P(J) \quad (7)$$

where $P(J|C)$ is the forward transliteration probability, $P(\acute{C}|J)$ is the backward transliteration probability, and $I(\acute{C}, C)$ is a penalty function to penalize those cases where $\acute{C}$ is not equal to $C$, i.e., $C$ fails to be covered from $J$. $P(J)$ is a Japanese Katakana language model.

3. Translate those Japanese names which are not in *i*CJ to Chinese in the similar way as step 2.

$$C^* = \underset{C}{\mathrm{argmax}}\, P(C|J) * P(\acute{J}|C) * I(\acute{J}, J) * P(C) \quad (8)$$

Note that the outputs of step 2 and 3 do not overlap with each other. $P(C)$ is a Chinese character-based language model.

4. Merge the results of step 1, 2 and 3. Given the merged data, we can easily train a direct Chinese-Japanese transliteration model.

The core idea of Algorithm 3 lies in eqs. (7) and (8). Among the three strategies (system-based, model-based and synthetic data-based), the first and the third ones are transliteration model independent while the second one is not.

### 3.4 Comparison with Previous Work

Almost all previous work on machine transliteration focuses on direct transliteration or transliteration system combination. Only two recent work (Khapra *et al.*, 2010; Zhang *et al.*, 2010) touches on the issue of pivot transliteration. Khapra *et al.* (2010) proposes the system-based strategy and does extensively empirical study together with CRF model on Indic/Slavic/Semetic languages and English. Zhang *et al.* (2010) proposes the model-based strategy, but reporting very bad performance. To address the low performance issue of the model-based strategy, this paper proposes the joint alignment algorithm to optimize the source-pivot-target alignment directly, resulting in significant performance improvement. Moreover, the paper proposes a new synthetic data-based strategy for pivot-based machine transliteration.

Machine translation carries out similar pivot-based translation studies. 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) study the model-level pivot approach and explore how to leverage on rule-based translation results in pivot language to improve translation performance. Utiyama and Isahara (2007) compare 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 ($P(<s,t>_k | <s,t>_{k-1})$). This is the fundamental difference.

## 4 Experimental Results

### 4.1 Experimental Settings

| Language Pair | Training | Test |
|---|---|---|
| Chinese-English (CE) | 31,961 | 2,896 |
| English-Japanese (EJ) | 23,225 | 1,489 |
| Chinese-English-Japanese (CEJ, the intersection part of CE and EJ) | 10,071 | 1,030 |

Table 1. Statistics on the data set

We use the NEWS 2009 Chinese-English and English-Japanese benchmark data as our experimental data (Li *et al.*, 2009a). All of the names originate from Western names, i.e., no native Chinese and Japanese names are involved in this experiment. Considering the fact that those language pairs with English involved have the most training data, it is reasonable to select English as pivot language. Table 1 reports the statistics of all the experimental data. The Chinese-English-Japanese data is the intersection of the Chinese-English and English-Japanese data.

We compare different alignment algorithms on the DEV set (Li *et al.*, 2009a). Finally we use Pervouchine *et al.* (2009)'s alignment algorithm for Chinese-Japanese and Li *et al.* (2004)'s for Chinese-English and English-Japanese. 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-10 pivot transliteration results. For the evaluation matrix, to save space, we only use top-1 accuracy (ACC) (Li *et al.*, 2009a) to measure transliteration performance since other five evaluation matrix used at Li *et al.* (2009a) are reported to have great correlation with ACC.

### 4.2 Experimental Results

#### 4.2.1 Results of Direct Transliteration

| Language Pair | ACC |
|---|---|
| English-Chinese | 0.681049 |
| English-Japanese | 0.456755 |
| Chinese-English | 0.394490 |
| Japanese-English | 0.314970 |
| Chinese-Japanese | 0.288022 |
| Japanese-Chinese | 0.366559 |

Table 2. Performance of direct transliterations

Table 2 reports the performance of direct transliteration. The first two experiments (line 1-2) are part of the NEWS 2009 share tasks and the others are our additional experiments for our pivot studies. Comparison of the first two experimental results and the results reported at NEWS 2009 shows that we achieve comparable performance with their best-reported systems under the same conditions of using single system and orthographic features only. This indicates that our baseline represents the state-of-the-art performance. In

| Methods | Chinese-English-Japanese | Japanese-English-Chinese |
|---|---|---|
| **Baseline 1**: Independent alignment of Chinese-English and English-Japanese (Zhang *et al.*, 2010) | 0.065949 (5816/16989) | 0.043011(5816/16989) |
| **Baseline 2**: Linguistically heuristic-based re-construction of Chinese-English and English-Japanese alignment (Zhang *et al.*, 2010) | 0.282638 (26351/34812) | 0.378299 (26351/34812) |
| **Method 1**: Joint alignment on intersection part of Chinese-English and English-Japanese data (Ours) | 0.287360 (26432/34920) | 0.378796 (26432/34920) |
| **Method 2**: Joint alignment on entire data set (Ours) | **0.325367 (37437/48590)** | **0.440782 (37437/48590)** |

Table 4. Performance of model-based strategy (in ACC/# of unigram/# of bigram of the different transliteration models learned y the model-based strategy)

addition, we find that the backward transliteration (line 3-4) consistently performs worse than its corresponding forward transliteration (line 1-2). 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 languages. This makes the transliteration from English to C/J a many-to-few mapping issue and backward transliteration a few-to-many mapping issue. Therefore backward transliteration has more ambiguities and thus is more difficult. Moreover, due to the less available training data for the language pairs without English involved (Chinese/Japanese), the lowest two experiments (line 5-6) performs worse than the case with English involved . This observation motivates us the study using pivot language for machine transliteration.

### 4.2.2 Results of System-based Strategy

| Language Pair | ACC |
|---|---|
| Chinese-English-Japanese (**System**) | 0.324361 |
| Chinese-English (**Direct**) | 0.394490 |
| English-Japanese (**Direct**) | 0.456755 |
| Chinese-Japanese (**Direct**) | 0.288022 |
| Japanese-English-Chinese (**System**) | 0.445748 |
| Japanese-English (**Direct**) | 0.314970 |
| English-Chinese (**Direct**) | 0.681049 |
| Japanese-Chinese (**Direct**) | 0.366559 |

Table 3. Performance of system-based strategy

Table 3 reports the experimental results of system-based strategy. It confirms the previous observations (Khapra *et al.*, 2010; Zhang *et al.*, 2010).

- The system-based pivot strategy is very effective, achieving significant performance improvement over direct transliteration.
- Different from other pipeline methodologies, system-based pivot strategy does not suffer from the error propagation issue. Its ACC is significantly better than the product of the ACCs of the two individual systems.

The main reasons of the good performance reported at the above observations are due to the following reasons:

- The pivot approach is able to use large amount of source-pivot and pivot-target data.
- The nature of transliteration is a phonetic translation process. 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 information.
- The N-best accuracy of machine transliteration is very high (Li *et al.*, 2004; Zhang *et al.*, 2004). It means that in most cases the correct transliteration in pivot language can be found in the top-10 results and the other 9 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.

### 4.2.3 Results of Model-based Strategy

Table 4 reports the performance of model-based strategy with different alignment refinements, where we can find:

- Baseline 1 clearly shows that the model-based strategy performs extremely worse than the other three settings if we align the Chinese-English and English-Japanese data independently. The major reason attributes to the size mismatching of the English transliteration units between the two data sets (syllable level *vs*. phoneme or syllable level).
- The other three experiments demonstrate the effectiveness of the alignment refinements. However Baseline 2 is more heuristic while ours are more mathematically principled.
- Method 1 performs comparably with Baseline 2 even utilizing fewer training data.
- Method 2 achieves the best performance. It convincingly shows the effectiveness of the proposed joint optimization algorithm.
- Among all the models, Method 2 has the largest amounts of model parameters (# of unigram and bigram). From modeling viewpoint, it indicates that this model is more powerful than others. This is due to the contribution of the more consistent English transliteration units.
- Comparing Tables 4 and Table 3, we can see that the model-based strategy performs as well as the system-based strategy. This clearly demonstrates the effectiveness of our proposed joint alignment algorithm.

### 4.2.4 Results of Synthetic Data-based Strategy

| Language Pair | ACC |
|---|---|
| Japanese-Chinese (**Synthetic Data**) | 0.465648 |
| Japanese-Chinese (**System**) | 0.445748 |
| Japanese-Chinese (**Model**) | 0.440782 |
| Japanese-Chinese (**Direct**) | 0.366559 |
| Chinese-Japanese (**Synthetic Data**) | 0.338930 |
| Chinese-Japanese (**System**) | 0.324361 |
| Chinese-Japanese (**Model**) | 0.325367 |
| Chinese-Japanese (**Direct**) | 0.288022 |

Table 5. Performance comparison of the three pivot strategies

Table 5 shows the advantage of synthetic data-based strategy over the other methods.

- The synthetic data-based strategy significantly outperforms the direct one. This clearly shows the effectiveness of the additional synthetic data.
- Using the same amount of data, the synthetic data-based strategy significantly outperforms the model-based one. This is not surprising since model-based strategy suffers from the transliteration unit mismatching issue and its performance is also compromised by the independent assumption of eq. (4) while the synthetic data-based directly learns the model from bilingual data without suffering from the above two issues.
- Using the same amount of data, the synthetic data-based strategy significantly outperforms the system-based one. This is because the synthetic data-based strategy directly learns a source-target transliteration model while system-based method utilizes two indirect models and has to bear with the independent assumption of eq. (2).
- It is worth noting the transliteration performance of the synthetic data-based strategy highly depends on the quality of the artificially generated data. Table 5 reports the performance using the default setting of eq. (7) and (8) at Algorithm 3. We expect that the synthetic data-based strategy has the potential to further improve its performance by simply introducing more features into eq. (7) and (8).

## 5 Conclusions

A big challenge to statistical-based machine transliteration is the lack of the training data, esp. to those low-density language pairs without English involved. To address the above issue, this paper propose a simple, but very effective solution, namely synthetic data-based strategy, to artificially generate direct source-target training data using pivot language. Experimental results on NEWS 2009 data shows that the proposed strategy is very useful, achieving the best-reported performance. The paper also proposes a joint alignment algorithm to jointly optimize the alignments between source, pivot and target data. Experimental results show that the joint alignment algorithm is able to largely boost the performance of model-based strategy.

The system-based and the proposed synthetic-based strategy are transliteration model-independent while model-based strategy is not. However, the three strategies and the proposed joint alignment algorithm are not limited to the machine transliteration task. They can be applied to those tasks which possess the similar "transitive" property as machine transliteration, such as paraphrasing, domain adaptation and some multi-lingual tasks.

## 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

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

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

D. Demner-Fushman and D. W. Oard. 2002. *The effect of bilingual term list size on dictionary-based cross-language information retrieval*. The 36th Hawaii Int'l. Conf. System Sciences

A.P. Dempster, 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

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

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

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

A. 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

H. Li, M. Zhang and J. 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

H. Meng, W. Lo, B. Chen and K. Tang. 2001. *Generate Phonetic Cognates to Handle Name Entities in English-Chinese cross-language spoken document retrieval*. ASRU-01

Jong-Hoon Oh and Key-Sun Choi. 2002. *An English-Korean Transliteration Model Using Pronunciation and Contextual Rules*. COLING-02

Jong-Hoon Oh, Kiyotaka Uchimoto, and Kentaro 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

R. Udupa, K. Saravanan, A. Bakalov and A. Bhole. 2009. *They are out there, if you know where to look: Mining transliterations of OOV query terms for cross-language information retrieval*. In LNCS: Advances in Information Retrieval, volume 5478, pages 437–448. Springer

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

H. Wu and H. Wang. 2007. *Pivot Language Approach for Phrase-based SMT*. ACL-07

H. Wu and H. 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

Min Zhang, Xiangyu Duan, Vladimir Pervouchine and Haizhou Li. 2010. *Machine Transliteration: Leveraging on Third Languages*. COLING-10 (poster)