An Effective and Robust Framework for Transliteration Exploration

Ea-Ee Jan, Niyu Ge
IBM T.J. Watson Research Center
Yorktown Height, NY 10598
{ejan,niyuge}@us.ibm.com

Shih-Hsiang Lin, Berlin Chen[#]
National Taiwan Normal University
Taipei, Taiwan
{shlin,berlin}@csie.ntnu.edu.tw

Abstract

Transliteration is the process of proper name translation based on pronunciation. It is an important process in many multilingual natural language tasks. A common and essential component of transliteration approaches is a verification mechanism that tests if the two names in different languages are translations of each other. Although many transliteration systems have verification as a component, verification as a stand-alone problem is relatively new. In this paper, we propose a simple, effective and robust training framework for the task of verification. We show the many applications of the verification techniques. Our proposed method can operate on both phonemic and orthographic inputs. Our best results show that a simple, straightforward orthographic representation is sufficient and no complex training method is needed. It is effective because it achieves remarkable accuracies. It is robust because it is language-independent. We show that on Chinese and Korean our technique achieves equal error rate well below 1% and around 1% for Japanese using 2009 and 2010 NEWS transliteration generation share task dataset. Our results also show that the orthographic system outperforms the phonemic system. This is especially encouraging because the orthographic inputs are easier to generate and secondly, one does not need to resort to more complex training algorithm to achieve excellent results. This approach is integrated for proper name based cross lingual information retrieval without translation.

1 Introduction

Proper name transliteration is important in many multilingual natural language processing tasks, such as Machine Translation (MT), Cross Lingual Information Retrieval (CLIR), multilingual spoken document retrieval and transliteration mining. The research community has investigated automatic proper name transliteration generation. The best performance with 10 references is approximately 70% for alphabet based edit distance error (Li et al., 2009). With one reference, the error rate can be as high as 50% (Meng et al., 2001; Virga and Khudanpur, 2003). If the error rate is measured using the whole proper name as a unit, the error rate will be even higher.

Alternatively, method for transliteration verification starts to draw attention in the research community. Given a pair of proper names in the source and target languages, the task is to decide whether they are transliterations of each other. This task is important for many applications. For example, in word alignment (Ittycheriah and Roukos, 2005), the unknown words are handled by computing a similarity score with the words in the target language. A similarity score derived from transliteration verification has been successfully applied to CLIR (Jan et. al., 2010). In their approach, CLIR can be achieved without translation of input proper name queries. More importantly, this technique is extremely useful in creating proper name pair training data (Kumaran et al., 2010). Given the vast amount of comparable data on the Internet, a technique that can reliably identify name pairs in different language is indispensable. (Kumaran et al., 2010) launched a new NEWS Transliteration Mining task. This task depends heavily on the accuracy of proper name verification techniques. In this paper, we propose a framework for the problem of transliteration verification. We show a highly accurate scoring mechanism that achieves very impressive results. This mechanism can be used as a tool for screening the transliteration par-

[#] This work was partially sponsored by "Aim for the Top University Plan" of National Taiwan Normal University and Ministry of Education, Taiwan.

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1332–1340,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

allel corpus, validating good data and filtering out bad data. In addition to the applications mentioned above, our method can also be used as an evaluation metric. The research community has been using methods such as word error rate, EER, precision and recall and its many variants as metrics to evaluate systems. However, due to homonyms and phone-set differences across multiple languages, word error rate is not always sufficient to distinguish transliteration accuracy. We envision our method as a novel and reliable metric in evaluating transliteration systems. Its simplicity, accuracy, and robustness will serve well as an automatic metric.

2 Background and Related Work

The problem of name transliteration was previously viewed as a translation problem. Virga and Khudanpur (2003) applied SMT models to translate English names into Chinese characters. Knight and Graehl (1997) proposed a generative transliteration model for Japanese and English using finite state transducers. Meng et al. (2001) developed an English-Chinese Named Entity transliteration technique using pronunciation lexicon and phonetic mapping rules. Li et al. (2004) proposed direct orthographic mapping with a joint source-channel model for proper name transliteration.

There have also been other approaches to transliteration. Al-Onaizan and Knight (2002) used verification as a stepping stone to transliteration. More recently, the JHU Workshop (2008) reported on the importance of the similarity scoring method and conducted a comparative study on the various scoring methods for name transliterations.

Data harvesting is another way of improving transliteration. Additional data source such as comparable corpora (Klementiev and Roth, 2006; Kuo et al., 2007; Sproat et al., 2006) and the web (Jiang et al., 2007) have also been explored to improve the performance. One of the vital building blocks in all of these approaches is a scoring component that tests how likely a given pair of names in source and target languages is transliteration of each other. This is a key component and is the aspect we focus on in this work. We propose a method for transliteration verification that achieves the best EER compared to other approaches on the same dataset.

Our work differentiates itself from the previous work in the following areas. We take the verification as a stand-alone problem the solution of which has a variety of NLP applications. We tackle the problem by using highly accurate and robust techniques. The verification task can be cast into an alignment problem. We use a generative model for alignment which renders similarity relationships between the source and target name pairs in phone sequences. In phoneme-based systems where phoneme generation might be ambiguous and error prone, we show a discriminative training method together with an HMM-based decoding strategy that works remarkably well within the framework. In orthographic systems where the input can be reliably generated, we show that the HMM-based strategy is sufficient. Section 3 presents our novel approach to verification. Section 4 and 5 show experiments and results. Section 6 and 7 demonstrate an application of our approach and future work.

3 A Highly Reliable Similarity Score

Transliteration between English and foreign language, especially Asian languages: e.g. Chinese, remains a big challenge. We investigate ways of using verification techniques for transliteration. To that end, we need a high quality verification mechanism. For a given proper name pair, one from source language and the other from target language, we want to verify with high precision if this pair refers to the same proper name. Our goal is to devise a scoring method that yields high accuracy with low computational complexity.

Intuitively, proper name transliteration "translates" a proper name based on pronunciation. For a pair of foreign name w_f, and English name w_e, the similarity can be defined as:

$$Sim(w_f, w_e) \cong Sim(ph_f, ph_e), \qquad (1)$$

where ph_f and ph_e are the corresponding phonetic sequences for the English and foreign names, respectively. Eq. (1) can be formulated as

$$\begin{aligned} Sim(ph_f, ph_e) = \lambda P(ph_f \mid \Lambda_{ph_e}) \\ + (1-\lambda) P(ph_e \mid \Lambda_{ph_f}) \end{aligned} \qquad (2)$$

where Λ_{ph_e} and Λ_{ph_f} are the English and foreign phonetic models, respectively. For simplification, it can be assumed that $\lambda = 0.5$ since the similarity function could be symmetric. Because the distributions of $P(ph_f \mid \Lambda_{ph_e})$ and $P(ph_e \mid \Lambda_{ph_f})$ are unknown, they need to be estimated through learning. Section 3.1 details the discriminative training process and section 3.2 presents an HMM-based decoding strategies to find the optimal alignment between $ph_f \; and \; ph_e$.

3.1 Model Estimation via SMT

One straightforward way to estimate the model parameters is to utilize the phrase tables produced by a phrase-based SMT framework. The phrase tables contain conditional probabilities of both *p(e|f)* and *p(f|e)*, which are the probabilities of English phrase given by foreign phrase and foreign phrase given by English phrase, respectively. When the phonetic sequences (either phonemic or orthographic) of English and foreign name pairs are the input into the SMT, the "phrase" table contains the phone set mappings between English and foreign phone sets together with their probabilities. We use these probabilities as the observation model in our HMM. We refer to this model as M_{SMT}.

3.2 Model Estimation via Discriminative Training

The discriminative training process involves finding an initial seed model and training in a decision-feedback learning framework.

One straightforward way to get an initial estimation for $P(ph_f \mid \Lambda_{ph_e})$ and $P(ph_e \mid \Lambda_{ph_f})$ is to utilize the phrase tables produced by the widely used phrase-based SMT system. The phrase tables contain both conditional probabilities of p(e|f) and p(f|e), which are the probabilities of English phrase given by foreign phrase and foreign phrase given by English phrase, respectively. When the phonetic sequences of English and foreign name pairs are fed into SMT, the "phrase" table contains the phone set mappings between English and foreign phone sets together with their probabilities. The phone set mapping is now data driven, and is free from the expensive and less flexible hand crafted linguistic phone set mapping rules. We refer to this model as M_{SMT}.

M_{SMT} is a straightforward and effective way to estimate the model parameters. Phoneme-based systems rely on the input texts being correctly converted to baseforms (phonemic sequences) representation. This process could be ambiguous, context-dependent, and error prone. In such systems, M_{SMT} serves as a good initial model. The model parameters can be further improved in a decision feed-back learning framework. The minimum classification error (MCE) training algorithm widely used in speech recognition can be applied here to improve the discrimination of the translation probability. We call this model M_{MCE}. Given a correct transliteration pair and other competitive transliteration hypotheses, we can define the transliteration error function as:

$$d_i(ph_f \mid \Lambda_{\mathbf{P}_e}) = -P(ph_f \mid \Lambda_{ph_e}) + \max_{f', f' \neq f} P(ph_{f'} \mid \Lambda_{ph_e}) \tag{3}$$

where $P(ph_f \mid \Lambda_{ph_e})$ is the alignment score obtained from the correct transliteration pair and $\max_{f', f' \neq f} P(ph_{f'} \mid \Lambda_{ph_e})$ is the highest competing score obtained from error transliteration pairs. The transliteration error function can be further transformed to a loss function ranging from 0 to 1 with the sigmoid operator:

$$l(d_i(ph_f \mid \Lambda_{ph_e})) = \frac{1}{1 + e^{(-\gamma d_i(ph_f \mid \Lambda_{ph_e}) + \theta)}} \tag{4}$$

where γ is used to control the slope of the function and θ is an offset factor. Above equation was then applied iteratively to update the translation probability:

$$p^{t+1}(ph_f \mid ph_e) = p^t(ph_f \mid ph_e) - \varepsilon \frac{\partial l(d_i(ph_f \mid \Lambda_{\mathbf{P}_e}))}{\partial p(ph_f \mid ph_e)} \tag{5}$$

3. 2 Decoding: Similarity Score Calculation

In order to calculate the similarity score for a given proper name pair (w_f, w_e), their respective phonetic sequence (ph_f, ph_e) is first determined. Then, for this task, we employ an HMM-based decoding strategy. The models $P(ph_f \mid \Lambda_{ph_e})$ and $P(ph_e \mid \Lambda_{ph_f})$ learned in section 3.1 are used as observation models. Two monotonic HMM models (one with ph_f as states and one with ph_e

as states) are then used to align the phonetic sequences according to Eq. (6) below:

$$P^*(ph_f \mid \Lambda_{ph_e}) = \arg\max_{\mathbf{S}_e} P(ph_f, \mathbf{S}_e \mid \Lambda_{ph_e}),$$
$$P^*(ph_e \mid \Lambda_{ph_f}) = \arg\max_{\mathbf{S}_f} P(ph_e, \mathbf{S}_f \mid \Lambda_{ph_f}) \quad (6)$$

where S_e is the English state sequence and S_f is the foreign state sequence.

The state transition probabilities are set to be uniform. We extend the traditional HMM to allow a broader range of phone mapping configurations. Specifically, the null transition (Bahl et al., 1982) is used to represent skipping a state without consuming any observations. This allows one to null mapping. The null state is introduced so it can emit those observations without any correspondence states. This allows null to one mapping. The combination of null transition and null state allow many to many and many to one configurations as well. The valid state transition is constrained to be from left to right with self loop, and with maximum jump of three states as well as a null state and a null transition.

Figure 1 depicts the actions of the HMM trellis at decode time. In Figure 1, the x-axis represents the observations (foreign language) and the y-axis represents the states (English). Take for example, the circle where dashed lines with arrows are emanating from. When this circle makes a horizontal move (from ph_{f2} to ph_{f3}), the single state ph_{e2} produces multiple observations. Null transition happens when the shaded circle makes a vertical move (from ph_{e2} to ph_{e3}) without consuming any observation.

4 Experiment setup for transliteration similarity

We evaluate the performance of our similarity scoring mechanism on 3 language pairs, Chinese-English (CE), Korean-English (KE), and Japanese-English (JE). Both Type I errors (false reject of the matched pairs) and Type II errors (false accept of the unmatched pairs) are evaluated. The Equal Error Rate (EER) is used as the evaluation metric.

For Chinese-English, a parallel corpus of proper name pairs is extracted from the *people* section of the multilingual Wikipedia. Among these, approximately 3,000 pairs are used for training and 300 pairs for testing. The 300 pairs are used as a matched condition test. A separate 1000 unmatched test pairs are created randomly from the 300 matched pairs.

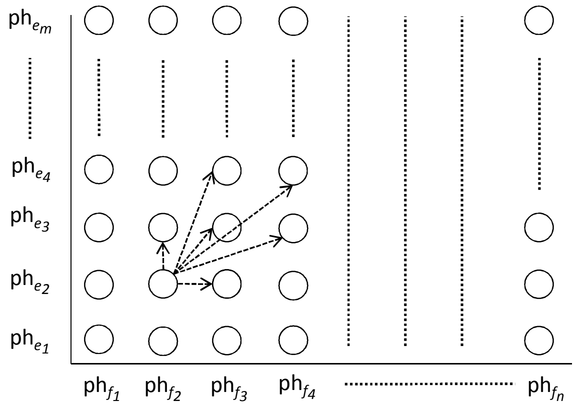

Figure 1 HMM Trellis

We also use the 2009 and 2010 NEWS transliteration generation shared task data as our test data. Although test our objective is different from those in the shared task, we choose this data because it is publicly available and can be used in the future for fair comparisons. We did not use NEWS 2010 transliteration miming shared task dataset because it did not contain Korean or Japanese. For Chinese, the 2009 data consists of 30K training and 2896 testing proper name pairs. Three systems are developed using 30K, 3K and 1K pairs of training data for our experiments. The 2896 proper name test pairs are used as matched pairs. Three unmatched test set pairs of size 10k, 100k and 1M are randomly generated. A 9M (2985x2986) unmatched pairs are also generated as an extreme test condition.

The Korean-English data comes from the 2010 NEWS transliteration generation data. It consists of 4,785 training pairs and 1,082 test pairs. Two systems with 1K and 4K of training pairs are developed; three sets unmatched pairs of size 10K, 100K, and 1M are generated. The Katakana Japanese-English data is from the same set (2010 NEWS data). It is bigger than the Korean data with 28K training instances and 1941 test pairs. Three systems with 1K, 4K and 28K training pairs are developed; three sets of unmatched pairs of size 10K, 100K, and 1M are also generated.

Training on 1K data matches the 2010 NEWS transliteration miming shared task (Kumaran et al., 2010) seed condition. Training on 3K-4K data matches the Wikipedia condition. Training on 28k for Japanese-English and 30K on Chinese-English demonstrates the best performance we can achieve while using all of the available training corpus.

We experiment with both phonemic and orthographic representations of input texts. The phonemic approach seems more intuitive since the transliteration is a pronunciation based translation. The orthographic system is simple because it does not require additional baseform generation tools to convert proper name to phonemic sequences, and it does not need to address the multiple pronunciation issue. For Chinese, the orthographic form of a character is its Pinyin. Tones in Pinyin are removed. Korean characters are converted according to Romanization tables from the web[1] . Japanese characters are Romanized in the same way using a different table[2] . We add 11 additional rules to the Japanese conversion process to deal with short versions of a few vowels and consonants. These 11 characters are: ァ, ィ, ゥ, ェ, ォ, ャ, ュ, ョ, ヮ, ー, and ッ. In orthographic systems, the Pinyin (for Chinese), Romanized spellings (for Korean and Japanese), and word spellings (for English) are then segmented into space delimited alphabet streams. For example, the English word ‘Clinton’ is segmented into seven letters separated by space ‘c l i n t o n’. In phoneme-based systems, diphthongs (such as ‘oi’, ‘ae’) and compound constants (such as ‘sh’) are treated as one unit. The English and Chinese baseforms are generated automatically from a speech recognition vendor toolkit. Multiple pronunciations for a given word are considered uniformly distributed. All possible combinations of pronunciation are created in both the training and the testing sets. All possible pronunciation combinations are used for training. The best score for all possible pronunciation combinations for a given proper pair is used for final score in testing.

In addition to the new approach described in section 3, we also build two phrase-based SMT systems, orthographic and phonemic based approach, for the Chinese-English Wikipedia datasets as a baseline. This SMT approach has been widely used and yields solid performance in shared task (Li et al, 2009, 2010). Equation (1) is reformulated as:

$$Sim(w_e, w_f) \cong Sim(tr(w_e), w_f) \approx BLEU(tr(w_e), w_f) \quad (7)$$

Model	EER
Orthographic edit distance	22%
Alphabet-based Orthographic SMT	6.47%
Phonetic SMT	7.10%
Our framework with M_{SMT}	3.73%
Our framework with M_{MCE}	3.33%

Table 1. CE Wikipedia Results with Baseline

Test	1K-Training			3K-Training		
	M_{SMT}	M_{MCE}	change	M_{SMT}	M_{MCE}	change
10K	1.37	1.27	7.06%	1.15	1.09	5.15%
100K	1.35	1.25	7.65%	1.17	1.11	5.52%
1M	1.39	1.26	9.09%	1.18	1.13	5.05%
9M	1.38	1.26	8.86%	1.18	1.12	5.23%
	30K-Training					
10k	1.07	1.02	4.63%			
100k	1.11	0.99	10.42%			
1M	1.17	1.00	14.47%			
9M	1.16	0.99	14.6%			

Table 2. CE 2009 NEWS Data

Test	1K-Training			4K-Training		
	M_{SMT}	M_{MCE}	Change	M_{SMT}	M_{MCE}	Change
10K	1.23	1.12	9.00%	1.12	0.99	10.79%
100K	1.21	1.16	4.03%	1.10	1.02	7.96%
1M	1.20	1.13	5.85%	1.09	1.00	8.67%

Table 3. KE 2010 NEWS Data

Test	1K-Training			4K-Training		
	M_{SMT}	M_{MCE}	change	M_{SMT}	M_{MCE}	change
10K	2.33	2.11	9.44%	2.09	2.02	3.35%
100K	2.40	2.19	8.75%	2.07	2.07	-
1M	2.40	2.19	8.75%	2.09	2.08	0.48%
	28K-Training					
10k	1.77	1.71	3.39%			
100k	1.76	1.70	3.41%			
1M	1.76	1.71	2.84%			

Table 4. JE 2010 NEWS Data

where tr(w_e) is the translation of w_e.

We chose BLEU (Papeneni et al., 2001) because it is more favorable to n-gram matches and is smoother than edit distance. We build a phonetic-based SMT and an alphabet orthographic-based SMT. In the former, the parallel data is converted to phonetic sequences using its own phone set. In

[1] The Korean Romanization table is from www.thelapan.com .

[2] The Japanese Romanization table is from http://www.omniglot.com/writing/japanese_katakana.htm

the orthographic SMT, the proper names are converted to their Pinyin in spelling form. The English proper names are put into spelling form as well. The standard SMT training recipe is then applied.

5 Results and discussions

The CE Wikipedia results are shown in Table 1. Our method with model M_{SMT} outperforms the traditional SMT methods and the orthographic edit distance approach. Our M_{MCE} further reduces the EER and achieves the best EER of 3.33%. This low EER shows that our verification approach is highly reliable.

Phoneme-based results on the NEWS data are shown in Tables 2, 3 and 4 for CE, KE, and JE respectively. Each table shows results of M_{SMT}, M_{MCE} and relative improvement (in that order) under different training and test conditions. From table 2, our approach yields less than 1.4% of EER using only 1K training pairs. Using 3K training data, the proposed method achieves ERR under 1.2%, which is comparable to the system using 30K training pairs. The MCE can further improve the performance relatively by 5-14%. In additions, the performance is very stable against to all different unmatched test conditions, especially at the 9M unmatched test pair condition.

The Japanese-English set performs worse than either Chinese or Korean. Upon inspection of the data, we find that the majority of the problems are due to incorrect baseforms representations. This, in turn, is because the Japanese data contains more non-English names. For example, in JE test set, there are 1941 matched pairs. For a 2% false reject rate, approximately 38 matched pairs are false rejected. Out of these false-reject entries, about a third is European names. Table 5 shows a few such examples. The bottom two entries in this table are actually incorrect transliteration pairs, which means they should be rejected but the system is penalized because the reference truth is not entirely clean. This is an example of using our method as a data screening tool to sift through the data and automatically pick out suspicious pairs. Because of our high accuracies, those questionable pairs can be either reliably excluded or down-weighted. They can also be given to annotators for further inspection. Instead of scanning through the entire dataset, human annotators can focus on just the disputable pairs that the system picks out. This annotation process is both efficient and cost-effective.

Japanese Katakana	**English**	**Romanized Japanese**
レーブ	Low	r e_ b u
ビュデ	Bade	b y u d e
ズバー	Zwar	z u b a_
ムジェール	Mjor	m u j e_ r u
ベア	Beer	b e a
ベーア	Bar	b e_ a
ミロスラフ	Cipar	m i r o s u r a f u
チャーチ	Chruch	ch a_ ch i

Table 5. JE problematic pair examples

Test	**1K-Training**	**3K-Training**	**30K-Training**
10K	0.87	0.73	0.58
100K	0.88	0.74	0.55
1M	0.87	0.73	0.56
9M	0.87	0.73	0.56

Table 6. $\mathbf{M_{SMT}}$ on orthographic CE

Test	**1K-Training**	**4K-Training**
10K	0.81%	0.74%
100K	0.83%	0.78%
1M	0.83%	0.79%

Table 7. $\mathbf{M_{SMT}}$ on orthographic KE

Test	**1K-Training**	**4K-Training**	**28K-Training**
10K	1.52%	0.97%	0.96%
100K	1.53%	1.05%	1.05%
1M	1.55%	1.04%	1.04%

Table 8. $\mathbf{M_{SMT}}$ on orthographic JE

Orthographic results are shown in Tables 6, 7, and 8 for CE, KE, and JE respectively. It is evident from the tables that orthographic-based systems are significantly better than the phoneme-based systems without using the more complex model $M_{MCE.}$ These results are very promising because first, orthographic representations do not need to deal with diphthongs and compound consonants. Every alphabet is a token by itself. In Table 5 for example, 'r e_ b u' in the first row will have '_' separated from 'e' in its orthographic form. Secondly, results in Tables 6, 7, and 8 are from systems using the straightforward SMT

Test	1K-Training		3K-Training	
	FR	FA	FR	FA
10K	0.79%	1.01%	0.76%	0.78%
100K		0.93%		0.73%
1M		0.56%		0.73%
9M		0.57%		0.73%
	30K-Training			
10K	0.59%	0.64%		
100K		0.55%		
1M		0.56%		
9M		0.57%		

Table 9. CE FR and FA rates

Test	1K-Training		4K-Training	
	FR	FA	FR	FA
10K	0.73%	1.07%	0.46%	1.13%
100K		1.04%		1.08%
1M		1.05%		1.09%

Table 10. KE FR and FA rates

Test	1K-Training		4K-Training	
	FR	FA	FR	FA
10K	1.80%	0.92%	1.13%	0.78%
100K		1.15%		1.01%
1M		1.16%		0.99%
	28K-Training			
10K	1.08%	0.73%		
100K		0.94%		
1M		0.93%		

Table 11. JE FR and FA rates

method without further discriminative training by *MCE*. This simplifies the overall system architecture and makes the system more efficient and effective.

One reason orthographic models perform better than phonemic models is that baseforms generation is ambiguous and error-prone. Our baseforms are statistically trained from a generic model. The conversion from input texts to their baseforms is a lossy process. The errors in Japanese show a clear example. When the names are non-English, the English baseforms all become incorrect which leads to verification errors. The orthographic representation alleviates this problem quite significantly and thus is able to improve the system. In addition to measuring ERR, we also measure False Rejection (FR) rate of the matched proper name pairs and False Acceptance (FA) rate of the unmatched pairs. Tables 9, 10, and 11 detail the results for all the language pairs under all testing and training conditions. For each language pair, under the same training condition, the FR rate is the same because given a fixed threshold, the number of matched pairs is the same.

FA and FR results in the above tables show that the system is very robust. Across all language pairs, FA and FR rates improve consistently as the training data size gets larger. The rates also remain stable across test data of different sizes.

6 Application

We incorporate the verification component into the retrieval model for CLIR. We use a language model (LM) based retrieval model. The query Q is treated as a sequence of words, $Q = w_1 w_2 \ldots w_N$. The query words are assumed to be independent of each other and conditionally independent given the document. The relevance score of a document to the query can be computed by Eq. (8):

$$P(Q|D) = \prod_{w_i \in Q} P(w_i|D)^{c(w_i,Q)}, \qquad (8)$$

where $c(w_i, Q)$ is the number of times that each distinct word w_i occurs in Q and $P(w_i|D)$ is the probability of the word w_i generated by the document model. For CLIR, we rewrite (8) as:

$$P(Q_e \mid D_f) = \prod_{w_e \in Q_e} P(w_e \mid D_f)^{c(w_e,Q_e)}$$
$$\overset{\text{rank}}{=} \sum_{w_e \in Q_e} c(w_e,Q_e) \log\left(\sum_{w_f \in D_f} P(w_e \mid w_f) P(w_f \mid D_f) \right) \qquad (9)$$

where the $P(w_e \mid w_f)$ is the probability of the English token given by foreign token. We propose to estimate this probability by a combination function of similarity function and translation table.

$$P(w_e \mid w_f) = \lambda \delta_0(w_e, w_f) + (1-\lambda)\delta_1(w_e, w_f), \qquad (10)$$

where

$$\delta_0(w_e, w_f) = \frac{1}{1+\exp(-\gamma \cdot sim(w_e, w_f) + \beta)},$$
$$\delta_1(w_e, w_f) = f(Tr(w_f \mid w_e))$$
$$f(Tr(w_f \mid w_e)) =$$
$$\frac{1}{1+\exp(-\gamma_1 \cdot sim(w_e, w_f) - \gamma_2 P(w_f \mid w_e) + \beta)}, \qquad (11)$$

Where, $sim(w_e, w_f)$ is the similarity function discussed in previous section, $Tr(w_e \mid w_f)$ is translations from the SMT phrase table, and f is a function to validate the phrase table entries. f is a function that combines the scores of the valid entries in the phrase table, $p(w_f \mid w_e)$, and the similarity score, $\delta_0(w_e, w_f)$, with higher recall rate. Thus, the entries in the phrase table with high scores are the candidates. These candidates will be discarded only if they are incorrect in pronunciation. The λ is the weighted factor for transliteration similarity scores, which can be a function of the total similarity scores. In our experiments, it is estimated by:

$$\lambda = \frac{k}{\sum_{w_f \in V_f} \delta_0(w_e, w_f)}, \qquad (13)$$

where v_f is the vocabulary and k is a constant.

We conduct experiments on the NTCIR-7 Information Retrieval for QA (IR4QA) task (Sakai et al. 2008). We select 10 proper name query topics as the query set. To test CLIR with multiple transliterations, we need a document collection with controlled multiple transliterations. We create a homogenous name list for those proper names used in the test query topics and uniformly place those names into the original document collections. Thus, each proper name in the query is replaced by 4-5 different names with similar pronunciation. The baseline (unigram document LM with Dirichlet smoothing) performance using the original queries against the synthetic document collection is 0.18 (in mAP). Without any given transliterations, the mAP of our method is 0.406, substantially better than 0.18 if one transliteration is given. This is shown in Table 12 where $\lambda = 1$ and # of known translations = 0. (λ is defined in Eq. (10)). In Table 12, $\lambda = 1$ implies all transliterations are ignored. We then test when two, or all transliterations are given without using the transliteration similarity by setting $\lambda = 0$. The results are 0.3819 and 0.7268, respectively. The mAP of 0.7268 is better than the mAP of 0.6911 from the ad-hoc baseline. It implies that the original document collections already have multiple transliterations. We further assume that all transliterations are known and one additional incorrect transliteration is provided. By disabling the filtering capability in Eq (11), (i.e. $f(Tr(w_f \mid w_e)$ =1, when $p(w_f \mid w_e)$ exists), the performance is degraded to 0.55. Table 12 shows that our approach is, in contrast, quite robust and maintains the performance of 0.65. This scenario with one incorrect transliteration can be very common because the entries in phrase table can be very noisy. We also evaluate the effect of name entities. If the entire document vocabulary is used to calculate similarity score, (cf. Eq (10)), the mAP=0.40, which means this task is very difficult and the name entities do not help significantly because too many name entities are extracted from the document collection. In fact, the name entities extraction may not be necessary while using our approach because the similarity score can be calculated based on the document vocabulary.

# of known Transliterations	λ =0	λ = variable	λ =1
0	0	0.4062	0.4062
1	0.1983	0.42	0.4062
2	0.3819	0.48	0.4062
All	0.7268	0.65	0.4062
All plus 1 wrong	0.55	0.65	0.4062

Table 12: mAP under various expanded transliterations for proper name queries

7 Conclusion and Future Work

In this paper, we propose a simple and effective transliteration verification framework. On the 2009 and 2010 NEWS transliteration generation shared task data, we achieve EER well below 1% for Chinese and Korean, and around 1% for Japanese. These promising results show that verification can not only provide an alternative approach to transliteration but also can be reliably used for exploring name pairs from comparable data. We show how the method is used in CLIR applications. It can also serve as a parallel corpus screening tool to indentify possible incorrect name pairs. In addition, it can be used for post processing of transliteration generation by filtering out incorrect top-n hypothesis to improve performance. Moreover, this approach can be turned into an automatic transliteration evaluation tool for such task as the NEWS shared task. In the future, we will explore each of these possibilities.

References

Yaser Al-Onaizan and Kevin Knight, 2002. Translating named entities using monolingual and bilingual resources. In Proc. of ACL-02. Pages: 400-408.

Lalit Bahl, Frederick Jelinek and Robert Mercer, A Maximum likelihood approach to continuous speech recognition. IEEE Transaction on attern Analysis and achine Intelligence, vol. PAMI-5, No.2, 1983, Pages 179-190.

Peter F. Brown, Stephan A. Della Pietra, and Robert L. Mercer, 1993. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2):263–311.

Abraham Ittycheriah and Salim Roukos, 2005. A maximum entropy word aligner for Arabic English machine translation. In Proceedings of EMNLP, pages 96-103

JHU Workshop Report, Multilingual Spoken Term Detection: Finding and Testing New Pronunciations In the final report of JHU Workshop 2008

Ea-Ee Jan, Shih-Hsiang Lin and Berlin Chan, Transliteration Retrieval Model for Cross Lingual Information Retrieval, AIRS 2010, Springer Lecture Notess Computer Science 6458, page 183-192

Long Jiang, Ming Zhou, Lee-Feng Chien, and Cheng Niu, Named entity translation with web mining and transliteration, Proceedings of the 20th international joint conference on Artificial intelligence, p.1629-1634, January 06-12, 2007, Hyderabad, India

Alexandre Klementiev and Dan Roth 2006, Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In Proceedings of the ACL 2006, Pages. 817-824 (2006)

Kevin Knight and Jonathan Graehl, 1998. Machine transliteration. Computational Linguistics, 24(4), Pages. 599-612, 1998

Philipp Koehn, Franz Josef Och, and Daniel Marcu, 2003, "Statistical Phrase based Translation", in Proc. Of HLT/NAACL Pages. 48-54, 2003.

Kumaran A, Mitesh Khapra and Haizhou Li, Report of NEWS 2010 Trasliteration Mining Shared Task. In Proc of 2010 Names Entities Workshop, ACL 2010, pages 21-28

Jin-Shea Kuo, Haizhou Li, and Ying-Kuei Yang, 2007. A phonetic similarity model for automatic extraction of transliteration pairs. ACM Transactions on Asian Language Information Processing. 6(2) Article 6: 1-24

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang, 2009. Report on news 2009 machine transliteration shared task. In Proceedings of ACLIJCNLP 2009 Named Entities workshop. pages. 1-18, Singapore.

Haizhou Li, Kumaran A, Zhang M. and Pervouchine V. Report of NEWS 2010 Transliteration Generation Shared Task. In Proceedings of ACL2010 Nameed Entity Workshop, Pages 1-11

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In Proceedings of ACL-04, pages 159–166, Barcelona, Spain, July.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins, 2002, Text Classification using String Kernels, Journal of Machine Learning Research 2 . pages 419-444

A Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, 1997, 'The DET Curve in Assessment of Detection Task Performance'. In: Proc. Eurospeech '97. Rhodes, Greece, Pages. 1895–1898.

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, Proceeding of ASRU 2001

Franz Josef Och, Hermann Ney, 2003, A Systematic Comparison of Various Statistical Alignment Models, Computational Linguistics, 29(1), 19-51 (2003).

Kishore A. Papeneni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation, In Proceedings of the 40th Annual of the Association for Computational linguistics, pages 311–318, Philadelphia, USA.

Richard Sproat, Tao Tao, and ChengXiang Zhai, 2006 Named entity transliteration with comparable corpora. In Proceedings of the COLING/ACL 2006. Pages 73-80

Tetsuya Sakai, Noriko Kando, Chuan-Jie Lin, Teruko Mitamura, Hideki Shima, Donghong Ji, Kuang-Hua Chen, Eric Nyberg, 2008. Overview of the NTCIR-7 ACLIA IR4QA Task. In: NTCIR-7 Workshop Meeting, Pages. 77-114

Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-lingual information retrieval. In Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition, pages 57–64, Sapporo, Japan.