Multi-Engine Based Machine Transliteration

Jong-Hoon Oh and Hitoshi Isahara
Computational Linguistics Group,
National Institute of Information and Communications Technology (NICT)
3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
{rovellia, isahara}@nict.go.jp

Abstract

We propose a novel approach for improving machine transliteration performance: *validating hypothesis transliterations derived from multiple transliteration engines*. Through experiments, we have shown that multiple transliteration engines and hypothesis transliterations improve machine transliteration performance.

1 Introduction

Machine transliteration has received significant attention as a tool for supporting machine translation (MT) [1, 2] and cross-language information retrieval (CLIR) [3]. A variety of different paradigms for machine transliteration have been developed over the years: a *grapheme-based transliteration model* (GTM) [4, 5, 6, 7], a *phoneme-based transliteration model* (PTM) [1, 4], a *hybrid transliteration model* (HTM) [2, 8], and a *correspondence-based transliteration model* (CTM) [9]. These models are classified in terms of the information sources used for transliteration or the units that are transliterated. Because each transliteration model depends on a particular information source, each one produces transliterations with errors. Moreover, different transliteration models usually produce different errors and different transliterations. Based on this observation, it seems to be possible to improve transliteration performance by combining the various transliteration models into one machine transliteration system that combines the merits of the individual ones and suffers from few of their demerits. In this paper, we propose a novel approach for improving machine transliteration performance: *validating hypothesis transliterations derived from multiple transliteration engines*.

This paper is organized as follows. We describe a framework for multi-engine based machine transliteration in Section 2, and present our experiments in Section 3. Finally, we conclude the paper in Section 4.

2 A Framework for Multi-Engine Based Machine Transliteration

What we concern here is how well we can construct multiple transliteration engines, at least one of which can generate correct hypothesis transliterations and how well we can assign reliable confidence scores to each hypothesis transliteration, which determines whether we can effectively select correct hypothesis transliterations.

2.1 Generating Hypothesis Transliterations

We use multiple transliteration engines based on GTM, PTM, HTM, and CTM to produce transliteration hypotheses. GTM, PTM, and CTM can generally function alone as transliteration engines, while HTM depends on other transliteration models to estimate its parameters. Therefore, we call a transliteration engine based on GTM, PTM, or CTM a "single-model engine" and one based on HTM a "hybrid-model engine". We use seven transliteration engines. Three are *single-model engines* corresponding to GTM, PTM, and CTM, respectively. Four are *hybrid-model engines*: three of them correspond to HTM using two of GTM, PTM, and CTM and the other is based on HTM using all three.

Let SW be a source word, P_{SW} be the pronunciation of SW, T_{SW} be a target word corresponding to SW, and C_{SW} be a correspondence between SW and P_{SW}. P_{SW} and T_{SW} can be segmented into a series of sub-strings, each of which corresponds to a source grapheme. Then, we can write $SW = s_1^n$, $P_{SW} = p_1^n$, $T_{SW} = t_1^n$, and $C_{SW} = c_1^n$, where s_i, p_i, t_i, and $c_i = (s_i, p_i)$ represent the i^{th} source grapheme, source phonemes corresponding to s_i, target graphemes corresponding to s_i and p_i, and the correspondence between s_i and p_i, respectively. With this definition, GTM ($SW \rightarrow T_{SW}$), PTM ($SW \rightarrow P_{SW}$ and $P_{SW} \rightarrow T_{SW}$), and CTM ($SW \rightarrow P_{SW}$ and $C_{SW} \rightarrow T_{SW}$) can be represented as Eqs. (1), (2), and (3), respectively. Given the assumption that each transliteration model depends on the size of the context, k, Eqs. (1), (2), and (3) can be simplified into a series of products.

$$\begin{aligned} Pr_{\mathcal{G}} &= Pr_{\mathcal{G}}(T_{SW}|SW) = Pr(t_1^n|s_1^n) \\ &\approx \prod_i Pr(t_i|t_{i-k}^{i-1}, s_{i-k}^{i+k}) \end{aligned} \tag{1}$$

$$\begin{aligned} Pr_{\mathcal{P}} &= Pr_{\mathcal{P}}(T_{SW}|SW) \\ &= Pr(p_1^n|s_1^n) \times Pr(t_1^n|p_1^n) \\ &\approx \prod_i Pr(p_i|p_{i-k}^{i-1}, s_{i-k}^{i+k}) \times Pr(t_i|t_{i-k}^{i-1}, p_{i-k}^{i+k}) \end{aligned} \tag{2}$$

$$\begin{aligned} Pr_{\mathcal{C}} &= Pr_{\mathcal{C}}(T_{SW}|SW) \\ &= Pr(p_1^n|s_1^n) \times Pr(t_1^n|c_1^n) \\ &\approx \prod_i Pr(p_i|p_{i-k}^{i-1}, s_{i-k}^{i+k}) \times Pr(t_i|t_{i-k}^{i-1}, c_{i-k}^{i+k}) \end{aligned} \tag{3}$$

To estimate the probabilities in Eqs. (1), (2), and (3), we use the maximum entropy model [10]. Our basic philosophy in designing feature functions for the maximum entropy model is that the context information collocated with the unit of interest is important. Based on this philosophy, we designed feature functions with all possible

combinations of (s_{i-k}^{i+k}, p_{i-k}^{i+k}, c_{i-k}^{i+k}, and t_{i-k}^{i-1}). Generally, a conditional maximum entropy model is an exponential log-linear model that gives the conditional probability of event $ev =< te, he >$, as described in Eq. (4), where λ_i is a parameter to be estimated [10].

$$Pr(te|he) = \frac{exp(\sum_i \lambda_i f_i(he, te))}{\sum_{te} exp(\sum_i \lambda_i f_i(he, te))} \quad (4)$$

With Eq. (4) and feature functions, conditional probabilities in Eqs. (1), (2), and (3) can be estimated like $Pr(t_i|t_{i-k}^{i-1}, c_{i-k}^{i+k}) = Pr(te|he)$, because we can represent target events (te) and history events (he) as t_i and tuples (t_{i-k}^{i-1}, c_{i-k}^{i+k}), respectively. In the same way, we can represent $Pr(t_i|t_{i-k}^{i-1}, s_{i-k}^{i+k})$, $Pr(t_i|t_{i-k}^{i-1}, p_{i-k}^{i+k})$, and $Pr(p_i|p_{i-k}^{i-1}, s_{i-k}^{i+k})$ as $Pr(te|he)$ with their corresponding target events and history events.

$$Pr_{\mathcal{H}(\mathcal{G}+\mathcal{P})}(T_{SW}|SW) \quad (5)$$
$$= \alpha \times Pr_{\mathcal{P}} + (1-\alpha) \times Pr_{\mathcal{G}}$$
$$Pr_{\mathcal{H}(\mathcal{G}+\mathcal{C})}(T_{SW}|SW) \quad (6)$$
$$= \beta \times Pr_{\mathcal{C}} + (1-\beta) \times Pr_{\mathcal{G}}$$
$$Pr_{\mathcal{H}(\mathcal{P}+\mathcal{C})}(T_{SW}|SW) \quad (7)$$
$$= \gamma \times Pr_{\mathcal{C}} + (1-\gamma) \times Pr_{\mathcal{P}}$$
$$Pr_{\mathcal{H}(\mathcal{G}+\mathcal{P}+\mathcal{C})}(T_{SW}|SW) \quad (8)$$
$$= \delta_1 \times Pr_{\mathcal{G}} + \delta_2 \times Pr_{\mathcal{P}} + \delta_3 \times Pr_{\mathcal{C}}$$

Using the definition of HTM [2, 8], we can represent four hybrid-model engines in a straightforward manner — Eqs. (5), (6), (7), and (8), where $\delta_1 + \delta_2 + \delta_3 = 1$ and $0 < \alpha, \beta, \gamma, \delta_1, \delta_2, \delta_3 < 1$. Note that $\mathcal{H}(\mathcal{G}+\mathcal{P})$ can be interpreted as HTM based on GTM ($\mathcal{G}$) and PTM ($\mathcal{P}$), and other notations can be interpreted in the same way. We used the "**maximum entropy modeling toolkit**" [11] to estimate Eqs. (1)–(8).

For each transliteration engine, we produce n-best transliteration hypotheses. We then make a set of transliteration hypotheses comprising the n-best transliteration hypotheses produced by the seven transliteration engines.

2.2 Validating Hypothesis Transliterations

We propose a validation model for hypothesis transliterations and letting it choose which one is a correct hypothesis transliteration. Let $\mathcal{HT}$ be a set of hypothesis transliterations (or transliteration candidates) produced by the seven transliteration engines, ht_i be the i^{th} hypothesis transliteration in $\mathcal{HT}$, and s be the source language word resulting in $\mathcal{HT}$. Then $S_{\mathbb{VTM}(\mathcal{X})}(s, ht_i)$, our validation model, can be represented as Eq. (9). $S_{\mathbb{VTM}(\mathcal{X})}(s, ht_i)$ is composed of two models, *transliteration engine-based model* ($S_{\mathbb{TTM}}$) and *Web-based model* ($S_{\mathbb{WTM}(\mathcal{X})}$).

$$S_{\mathbb{VTM}(\mathcal{X})}(s, ht_i) = S_{\mathbb{TTM}}(s, ht_i) \times S_{\mathbb{WTM}(\mathcal{X})}(s, ht_i) \quad (9)$$

2.2.1 Transliteration Engine-based Model

$S_{\mathbb{TTM}}(s, ht_i)$ is based on the original rank assigned by each transliteration engine. For a given source word (s), each transliteration engine generates hypothesis transliterations and assigns their ranks with the probability described in Eqs. (1)–(3), and (5)–(8). The underlying assumption is that the rank of the correct transliterations tends to be higher, on average, than the wrong ones. Let τ be a ranked list of hypothesis transliterations produced by certain transliteration engine, $|\tau|$ be the number of hypothesis transliterations in τ, R be a set of the ranked lists ($|R| = 7$ in this paper), and $rank(\tau, i)$ be a rank of item (s, ht_i) ($rank(\tau, i) = 1$ indicates Top-1 of τ). If (s, ht_i) is not in the τ, we assign $rank(\tau, i) = |\tau| + 1$. Then we can represent $S_{\mathbb{TTM}}$ (Eq. (10)) with a rank normalization method [12].

$$S_{\mathbb{TTM}}(s, ht_i) = \frac{1}{|R|} \sum_{\tau \in R} (1 - \frac{rank(\tau, i) - 1}{|\tau|}) \quad (10)$$

2.2.2 Web-Based Model

Korean and Japanese Web pages are usually composed of rich texts in a mixture of Korean or Japanese (main language) and English (auxiliary language). Let s and t be a source language word and a target language word, respectively. We observed that s and t tend to be near each other in the text of Korean or Japanese Web pages when the authors of the Web pages describe s as translation of t, or vice versa. We retrieved such Web pages for Web-based confidence scores.

There have been several studies of translation validation or transliteration validation [2, 13, 14, 15] based on Web data. Generally they have relied on Web frequency (the number of Web pages retrieved by a Web search engine). The chi-square (χ^2) test and relative Web frequency derived from "**bilingual keyword search** (BKS)" [14, 15] or "**monolingual keyword search** (MKS)" [2, 13] have been used in their validation. BKS retrieves Web pages by using a query composed of two keywords, s and t, while MKS retrieves Web pages by using a query composed only of t. However, Web pages retrieved by MKS tend to show whether t is used in target language texts rather than whether t is a translation of s. BKS frequently retrieves Web pages where s and t have little relation to each other because it does not consider distance between s and t in the Web pages. To address these problems, we use "**bilingual phrasal search** (BPS)", where a phrase composed of s and t is used as a query for a search engine.

$$S_{\mathbb{WTM}(\mathcal{X})}(s, ht_i) = \frac{1 + WF_{\mathcal{X}}(s, ht_i)}{|\mathcal{HT}| + \sum_{ht_k \in \mathcal{HT}} WF_{\mathcal{X}}(s, ht_k)} \quad (11)$$

Let $WF_{\mathcal{X}}(s, ht_i)$ be the Web frequency retrieved by $\mathcal{X} \in \{MKS, BKS, BPS\}$. Then, $S_{\mathbb{WTM}(\mathcal{X})}$ can be represented as Eq. (11), which represents the relative Web frequency. To avoid zero value in $S_{\mathbb{WTM}(\mathcal{X})}$, we use the Laplace smoothing method [16].

		EKSet				EJSet			
		Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10
$GTM(\mathcal{G})$		56.8	75.5	80.3	84.4	51.6	71.8	77.2	80.9
$PTM(\mathcal{P})$		49.6	66.5	72.0	77.3	47.6	67.9	73.2	77.9
$CTM(\mathcal{C})$		60.8	78.8	83.1	86.6	58.2	79.0	84.7	89.7
$HTM_{(\mathcal{G}+\mathcal{P})}$		60.6	78.2	83.0	87.5	56.5	76.3	82.4	87.9
$HTM_{(\mathcal{G}+\mathcal{C})}$		62.1	80.2	84.9	88.9	59.0	80.4	87.3	94.1
$HTM_{(\mathcal{P}+\mathcal{C})}$		61.3	77.2	81.5	85.2	58.9	78.6	84.5	89.6
$HTM_{(\mathcal{G}+\mathcal{P}+\mathcal{C})}$		62.1	79.7	84.1	88.4	59.3	79.6	85.7	91.1
$S_{\mathbb{TM}}$		72.6	80.3	84.9	88.8	68.8	79.6	85.7	91.6
$S_{\mathbb{WM}}$	$S_{\mathbb{WM}(\mathcal{MKS})}$	25.1	46.2	59.4	77.9	36.4	61.3	73.7	88.3
	$S_{\mathbb{WM}(\mathcal{BKS})}$	66.8	85.2	89.1	91.4	69.2	87.1	91.7	95.0
	$S_{\mathbb{WM}(\mathcal{BPS})}$	84.8	91.0	91.4	91.7	79.3	92.4	93.9	95.1
$S_{\mathbb{VM}}$	$S_{\mathbb{TM}} \times S_{\mathbb{WM}(\mathcal{MKS})}$	31.3	54.0	67.3	83.0	43.6	68.9	80.3	91.0
	$S_{\mathbb{TM}} \times S_{\mathbb{WM}(\mathcal{BKS})}$	71.4	87.4	90.1	91.6	72.7	89.1	92.9	95.4
	$S_{\mathbb{TM}} \times S_{\mathbb{WM}(\mathcal{BPS})}$	85.3	91.7	91.9	92.0	80.5	93.4	94.5	95.2

Table 1: Summary of Results (%)

3 Evaluation

Our experiments were done with English-to-Korean and English-to-Japanese transliteration. The test set for the English-to-Korean transliteration (EKSet) [17] consisted of 7,172 English-Korean pairs. The test set for the English-to-Japanese transliteration (EJSet), which consisted of English-katakana pairs from EDICT[1], consisted of 10,417 pairs. EJSet contained one or more correct transliterations for each English word, for example, <*micro*, 'ma-i-ku-ro'>, and <*micro*, 'mi-ku-ro'>. The average number of Japanese transliterations for an English word was 1.15. EKSet and EJSet covered proper names, technical terms, and general terms. Evaluation was done in terms of word accuracy (WA) in Eq. (12), where TRLs means transliterations. In the evaluation, we used k-fold cross-validation ($k = 7$ for EKSet and $k = 10$ for EJSet). The test set was divided into k subsets. Each one was used for testing, while the remainder was used for training. Then, average WA across all k trials was computed. Through the cross-validation, we set the size of the context window for $Pr_{\mathcal{G}}$, $Pr_{\mathcal{P}}$, $Pr_{\mathcal{C}}$, and $Pr_{\mathcal{H}}$ in Eqs. (1)–(8) at 3, and linear interpolation parameters of hybrid-model engines at (α, β, γ, δ_1, δ_2, and δ_3)[2].

$$WA = \frac{correct\ TRLs.\ output\ by\ the\ system}{TRLs.\ in\ the\ blind\ test\ data} \quad (12)$$

3.1 Results

A summary of our experimental results conducted on EKSet and EJSet is shown in Table 1. In the experiment, each transliteration engine generated 10-best hypothesis transliterations. In the table, GTM, PTM, CTM, and the HTMs represent the individual transliteration models used for transliteration engines. $S_{\mathbb{TM}}$, $S_{\mathbb{WM}}$, and $S_{\mathbb{VM}}$ represent experimental results, where hypothesis transliterations generated by the seven transliteration engines are validated by Eqs. (10), (11), and (9), respectively. Moreover, we tested the effect of BPS, BKS, and MKS on $S_{\mathbb{WM}}$ and $S_{\mathbb{VM}}$. The Top-n considers whether the correct transliteration is in the Top-n ranked list[3].

Compared to individual transliteration engines, multiple transliteration engines equipped with $S_{\mathbb{TM}}$, $S_{\mathbb{WM}}$, and $S_{\mathbb{VM}}$ performed better, especially in the Top-1. $S_{\mathbb{TM}}$ by itself showed higher performance than any other individual transliteration engine. $S_{\mathbb{WM}}$ depends on the Web to investigate whether generated hypothesis transliterations are frequently used in real-world texts and $S_{\mathbb{WM}}$ selects the most relevant hypothesis transliterations through investigation. Based on Web search methods, $S_{\mathbb{WM}}$ showed different results. $S_{\mathbb{WM}(\mathcal{MKS})}$ has the worst performance because it tends to validate whether ht_i is used in a target language rather than whether it is used as a translation of its source language word (s). On the other hand, $S_{\mathbb{WM}(\mathcal{BKS})}$ and $S_{\mathbb{WM}(\mathcal{BPS})}$ are effective because BKS and BPS can consider both s and ht_i. Comparing $S_{\mathbb{WM}(\mathcal{BKS})}$ to $S_{\mathbb{WM}(\mathcal{BPS})}$, $S_{\mathbb{WM}(\mathcal{BPS})}$ showed higher accuracy in Top-1 because of the rigid Web search condition, *phrasal search*, in BPS. BPS has better ability to retrieve reliable Web pages for validating hypothesis transliterations than BKS. However, $\sum_{ht_k \in \mathcal{HT}} WF_{\mathcal{X}}(s, ht_k) = 0$ happens in BPS more often than in BKS. It can decrease the performance of $S_{\mathbb{WM}(\mathcal{BPS})}$ in validating hypothesis transliterations, because $S_{\mathbb{WM}(\mathcal{BPS})}$ will assign the same confidence score to all hypotheses in $\mathcal{HT}$. However, we find that the problem usually happens in BPS when $\mathcal{HT}$ does not include correct hypothesis transliterations; thus it does not signif-

[1] http://www.csse.monash.edu.au/~jwb/edict.html

[2] We set the parameters showing the highest Top-1 performance in the k-fold validation. $\alpha = 0.4$, $\beta = 0.5$, $\gamma = 0.7$, $\delta_1 = 0.4$, $\delta_2 = 0.2$, and $\delta_3 = 0.4$ for EKSet and $\alpha = 0.4$, $\beta = 0.5$, $\gamma = 0.7$, $\delta_1 = 0.2$, $\delta_2 = 0.2$, and $\delta_3 = 0.6$ for EJSet.

[3] For one English word, there are one or more correct transliterations in EJSet but there is only one correct transliteration in EKSet. Therefore, the individual models showed higher TOP-1 accuracies but lower TOP-10 accuracies in EKSet than in EJSet.

icantly degrade the performance of $S_{\mathrm{WIM}(\mathcal{BPS})}$.

Multiple transliteration engines provide more chances to find correct hypothesis transliterations by generating various hypothesis transliterations, and S_{VIM} assigns each hypothesis transliteration reliable confidence scores, with which we can effectively decide which one is a correct hypothesis transliteration. For these reasons, $S_{\mathrm{VIM}(\mathcal{BPS})}$ shows the best performance. More specifically, $S_{\mathrm{WIM}(\mathcal{BPS})}$ contributes a lot to the improved performance, while S_{TIM} contributes little to the improved performance because $S_{\mathrm{WIM}(\mathcal{BPS})}$ correctly validates transliterations whenever S_{TIM} does. However, S_{TIM} in S_{VIM} well compensated for the errors of S_{WIM} in MKS and BKS.

We compared our approach with several previous studies: Kang and Choi [4] (GTM and PTM), Kang and Kim [5] (GTM), Goto *et al.* [6] (GTM), Bilac and Tanaka [8] ($HTM_{(\mathcal{G}+\mathcal{P})}$), and Oh and Choi [9] ($CTM$), each of which corresponds to the transliteration model set off in parentheses. We implemented transliteration methods of these studies with the same training and test data as our proposed approach. We found that their performance was similar to that of the individual transliteration engines in Table 1 if they are based on the same transliteration model.

4 Conclusion

This paper describes a novel approach for improving machine transliteration performance based on a *validating hypothesis transliterations derived from multiple transliteration engines* strategy. We produced hypothesis transliterations using seven transliteration engines and validated them based on Web frequency of hypothesis transliteration and the rank of each hypothesis transliteration assigned by individual transliteration engines. Through experiments, we have shown that multiple transliteration engines and hypothesis transliterations improve machine transliteration performance. Moreover, we have shown that our transliteration validation model is effective in selecting correct hypothesis transliterations.

However, further research is needed to further improve performance. A more sophisticated S_{TIM} is necessary. Rank aggregation methods used in meta-search engines such as a Markov chain-based method [12] might be helpful in addressing the problem.

References

[1] K. Knight and J. Graehl. Machine transliteration. In *Proc. of the 35th Annual Meetings of the Association for Computational Linguistics*, pages pp.128–135, 1997.

[2] Y. Al-Onaizan and Kevin Knight. Translating named entities using monolingual and bilingual resources. In *Proc. of ACL 2002*, pages 400–408, 2002.

[3] Atsushi Fujii and Ishikawa Tetsuya. Japanese/English cross-language information retrieval: Exploration of query translation and transliteration. *Computers and the Humanities*, 35(4):389–420, 2001.

[4] B. J. Kang and K. S. Choi. Automatic transliteration and back-transliteration by decision tree learning. In *Proc. of LREC 2000*, pages 1135–1411, 2000.

[5] I. H. Kang and G. C. Kim. English-to-Korean transliteration using multiple unbounded overlapping phoneme chunks. In *Proc. of COLING 2000*, pages 418–424, 2000.

[6] I. Goto, N. Kato, N. Uratani, and T. Ehara. Transliteration considering context information based on the maximum entropy method. In *Proc. of MT-Summit IX*, pages 125–132, 2003.

[7] H. Li, M. Zhang, and J. Su. A joint source-channel model for machine transliteration. In *Proc. of ACL 2004*, pages 160–167, 2004.

[8] Slaven Bilac and Hozumi Tanaka. Direct combination of spelling and pronunciation information for robust back-transliteration. In *Proc. of CICLing 2005*, pages 413 – 424, 2005.

[9] J.-H. Oh and K.-S. Choi. An ensemble of grapheme and phoneme for machine transliteration. In *Proc. of IJCNLP05*, pages 450–461, 2005.

[10] A. L. Berger, S. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[11] L. Zhang. Maximum entropy modeling toolkit for python and C++. `http://homepages.inf.ed.ac.uk/s0450736/software/maxent/manual.pdf`, 2004.

[12] M. Elena Renda and Umberto Straccia. Web metasearch: Rank vs. score based rank aggregation methods. In *Proc. 18th Annual ACM Symposium on Applied Computing (SAC-3)*, pages 841–846, 2003.

[13] Gregory Grefenstette, Yan Qu, and David A. Evans. Mining the web to create a language model for mapping between English names and phrases and Japanese. In *Proc. of Web Intelligence*, pages 110–116, 2004.

[14] Yan Qu and Gregory Grefenstette. Finding ideographic representations of Japanese names written in Latin script via language identification and corpus validation. In *Proc. of ACL*, pages 183–190, 2004.

[15] Jenq-Haur Wang, Jei-Wen Teng, Wen-Hsiang Lu, and Lee-Feng Chien. Exploiting the web as the multilingual corpus for unknown query translation. *JASIST*, 57(5):660–670, 2006.

[16] C.D. Manning and Hinrich Schutze. *Foundations of Statistical natural language Processing*. MIT Press, 1999.

[17] Y. S. Nam. *Foreign dictionary*. Sung An Dang, 1997.