Latent Class Transliteration based on Source Language Origin

Masato Hagiwara
Rakuten Institute of Technology, New York
215 Park Avenue South, New York, NY
masato.hagiwara@mail.rakuten.com

Satoshi Sekine
Rakuten Institute of Technology, New York
215 Park Avenue South, New York, NY
satoshi.b.sekine@mail.rakuten.com

Abstract

Transliteration, a rich source of proper noun spelling variations, is usually recognized by phonetic- or spelling-based models. However, a single model cannot deal with different words from different language origins, e.g., "get" in "piaget" and "target." Li et al. (2007) propose a method which explicitly models and classifies the source language origins and switches transliteration models accordingly. This model, however, requires an explicitly tagged training set with language origins. We propose a novel method which models language origins as latent classes. The parameters are learned from a set of transliterated word pairs via the EM algorithm. The experimental results of the transliteration task of Western names to Japanese show that the proposed model can achieve higher accuracy compared to the conventional models without latent classes.

1 Introduction

Transliteration (e.g., "バラクオバマ *baraku obama* / Barak Obama") is phonetic translation between languages with different writing systems. Words are often transliterated when imported into differet languages, which is a major cause of spelling variations of proper nouns in Japanese and many other languages. Accurate transliteration is also the key to robust machine translation systems.

Phonetic-based rewriting models (Knight and Jonathan, 1998) and spelling-based supervised models (Brill and Moore, 2000) have been proposed for recognizing word-to-word transliteration correspondence. These methods usually learn a single model given a training set. However, single models cannot deal with words from multiple language origins. For example, the "get" parts in "piaget / ピアジェ *piaje*" (French origin) and "target / ターゲット *tāgetto*" (English origin) may differ in how they are transliterated depending on their origins.

Li et al. (2007) tackled this issue by proposing a *class transliteration model*, which explicitly models and classifies origins such as language and genders, and switches corresponding transliteration model. This method requires training sets of transliterated word pairs with language origin. However, it is difficult to obtain such tagged data, especially for proper nouns, a rich source of transliterated words. In addition, the explicitly tagged language origins are not necessarily helpful for loanwords. For example, the word "spaghetti" (Italian origin) can also be found in an English dictionary, but applying an English model can lead to unwanted results.

In this paper, we propose a *latent class transliteration model*, which models the source language origin as unobservable latent classes and applies appropriate transliteration models to given transliteration pairs. The model parameters are learned via the EM algorithm from training sets of transliterated pairs. We expect that, for example, a latent class which is mostly occupied by Italian words would be assigned to "spaghetti / スパゲティ *supageti*" and the pair will be correctly recognized.

In the evaluation experiments, we evaluated the accuracy in estimating a corresponding Japanese transliteration given an unknown foreign word,

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 53–57,
Portland, Oregon, June 19-24, 2011.

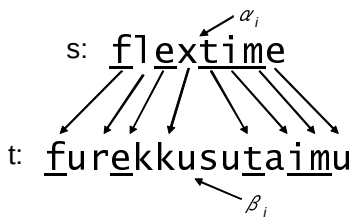

Figure 1: Minimum edit operation sequence in the alpha-beta model (Underlined letters are match operations)

using lists of Western names with mixed languages. The results showed that the proposed model achieves higher accuracy than conventional models without latent classes.

Related researches include Llitjos and Black (2001), where it is shown that source language origins may improve the pronunciation of proper nouns in text-to-speech systems. Another one by Ahmad and Kondrak (2005) estimates character-based error probabilities from query logs via the EM algorithm. This model is less general than ours because it only deals with character-based error probability.

2 Alpha-Beta Model

We adopted the *alpha-beta model* (Brill and Moore, 2000), which directly models the string substitution probabilities of transliterated pairs, as the base model in this paper. This model is an extension to the conventional edit distance, and gives probabilities to general string substitutions in the form of $\alpha \rightarrow \beta$ (α, β are strings of any length). The whole probability of rewriting word s with t is given by:

$$P_{AB}(t|s) = \max_{T \in \mathrm{Part}(t), S \in \mathrm{Part}(s)} \prod_{i=1}^{|S|} P(\alpha_i \rightarrow \beta_i), \quad (1)$$

where $\mathrm{Part}(x)$ is all the possible partitions of word x. Taking logarithm and regarding $-\log P(\alpha \rightarrow \beta)$ as the substitution cost of $\alpha \rightarrow \beta$, this maximization is equivalent to finding a minimum of total substitution costs, which can be solved by normal dynamic programming (DP). In practice, we conditioned $P(\alpha \rightarrow \beta)$ by the position of α in words, i.e., at the beginning, in the middle, or at the end of the word. This conditioning is simply omitted in the equations in this paper.

The substitution probabilities $P(\alpha \rightarrow \beta)$ are learned from transliterated pairs. Firstly, we obtain an edit operation sequence using the normal DP for edit distance computation. In Figure 1 the sequence is f→f, ε →u, l→r, e→e, ε →k, x→k, ... and so on. Secondly, non-match operations are merged with adjacent edit operations, with the maximum length of substitution pairs limited to W. When $W = 2$, for example, the first non-match operation ε →u is merged with one operation on the left and right, producing f→fu and l→ur. Finally, substitution probabilities are calculated as relative frequencies of all substitution operations created in this way. Note that the minimum edit operation sequence is not unique, so we take the averaged frequencies of all the possible minimum sequences.

3 Class Transliteration Model

The alpha-beta model showed better performance in tasks such as spelling correction (Brill and Moore, 2000), transliteration (Brill et al., 2001), and query alteration (Hagiwara and Suzuki, 2009). However, the substitution probabilities learned by this model are simply the monolithic average of training set statistics, and cannot be switched depending on the source language origin of given pairs, as explained in Section 1.

Li et al. (2007) pointed out that similar problems arise in Chinese. Transliteration of Indo-European names such as "亜歴山大 / Alexandra" can be addressed by Mandarin pronunciation (*Pinyin*) "*Ya-Li-Shan-Da*," while Japanese names such as "山本 / Yamamoto" can only be addressed by considering the Japanese pronunciation, not the Chinese pronunciation "*Shan-Ben*." Therefore, Li et al. took into consideration two additional factors, i.e., source language origin l and gender / first / last names g, and proposed a model which linearly combines the conditioned probabilities $P(t|s,l,g)$ to obtain the transliteration probability of $s \rightarrow t$ as:

$$\begin{aligned} P(t|s)_{\mathrm{soft}} &= \sum_{l,g} P(t,l,g|s) \\ &= \sum_{l,g} P(t|s,l,g) P(l,g|s) \quad (2) \end{aligned}$$

We call the factors $c = (l,g)$ as *classes* in this paper. This model can be interpreted as firstly computing

the class probability distribution given $P(c|s)$ then taking a weighted sum of $P(t|s,c)$ with regard to the estimated class c and the target t.

Note that this weighted sum can be regarded as doing *soft-clustering* of the input s into classes with probabilities. Alternatively, we can employ *hard-clustering* by taking one class such that $c^* = \arg\max_{l,g} P(l,g|s)$ and compute the transliteration probability by:

$$P(t|s)_{\text{hard}} \quad \propto \quad P(t|s,c^*). \tag{3}$$

4 Latent Class Transliteration Model

The model explained in the previous section integrates different transliteration models for words with different language origins, but it requires us to build class detection model c from training pairs explicitly tagged with language origins.

Instead of assigning an explicit class c to each transliterated pair, we can introduce a random variable z and consider a conditioned string substitution probability $P(\alpha \rightarrow \beta|z)$. This latent class z corresponds to the classes of transliterated pairs which share the same transliteration characteristics, such as language origins and genders. Although z is not directly observable from sets of transliterated words, we can compute it via EM algorithm so that it maximizes the training set likelihood as shown below. Due to the space limitation, we only show the update equations. X_{train} is the training set consisting of transliterated pairs $\{(s_n,t_n)|1 \leq n \leq N\}$, N is the number of training pairs, and K is the number of latent classes.

$$\textbf{Parameters:} \quad P(z=k)=\pi_k, P(\alpha \rightarrow \beta|z) \tag{4}$$

$$\textbf{E-Step:} \quad \gamma_{nk} = \frac{\pi_k P(t_n|s_n, z=k)}{\sum_{k=1}^{K} \pi_k P(t_n|s_n, z=k)}, \tag{5}$$

$$P(t_n|s_n,z) = \max_{T \in \text{Part}(t_n), S \in \text{Part}(s_n)} \prod_{i=1}^{|S|} P(\alpha_i \rightarrow \beta_i|z)$$

$$\textbf{M-Step:} \quad \pi_k^* = \frac{N_k}{N}, \quad N_k = \sum_{n=1}^{N} \gamma_{nk} \tag{6}$$

$$P(\alpha \rightarrow \beta|z=k)^* = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \frac{f_n(\alpha \rightarrow \beta)}{\sum_{\alpha \rightarrow \beta} f_n(\alpha \rightarrow \beta)}$$

Here, $f_n(\alpha \rightarrow \beta)$ is the frequency of substitution pair $\alpha \rightarrow \beta$ in the n-th transliterated pair, whose calculation method is explained in Section 2. The final transliteration probability is given by:

$$P_{\text{latent}}(t|s) = \sum_z P(t,z|s) = \sum_z P(z|s)P(t|s,z)$$
$$\propto \sum_z \pi_k P(s|z)P(t|s,z) \tag{7}$$

The proposed model cannot explicitly model $P(s|z)$, which is in practice approximated by $P(t|s,z)$. Even omitting this factor only has a marginal effect on the performance (within 1.1%).

5 Experiments

Here we evaluate the performance of the transliteration models as an information retrieval task, where the model ranks target t' for a given source s', based on the model $P(t'|s')$. We used all the t'_n in the test set $X_{test} = \{(s'_n, t'_n)|1 \leq n \leq M\}$ as target candidates and s'_n for queries. Five-fold cross validation was adopted when learning the models, that is, the datasets described in the next subsections are equally splitted into five folds, of which four were used for training and one for testing. The mean reciprocal rank (MRR) of top 10 ranked candidates was used as a performance measure.

5.1 Experimental Settings

Dataset 1: Western Person Name List This dataset contains 6,717 Western person names and their Katakana readings taken from an European name website 欧羅巴人名録 [1], consisting of German (de), English (en), and French (fr) person name pairs. The numbers of pairs for these languages are 2,470, 2,492, and 1,747, respectively. Accent marks for non-English languages were left untouched. Uppercase was normalized to lowercase.

Dataset 2: Western Proper Noun List This dataset contains 11,323 proper nouns and their Japanese counterparts extracted from Wikipedia interwiki. The languages and numbers of pairs contained are: German (de): 2,003, English (en): 5,530, Spanish (es): 781, French (fr): 1,918, Italian (it):

[1] http://www.worldsys.org/europe/

Language	de	en	fr
Precision(%)	80.4	77.1	74.7

Table 1: Language Class Detection Result (Dataset 1)

Language	de	en	es	fr	it
Precision(%)	65.4	83.3	48.2	57.7	66.1

Table 2: Language Class Detection Result (Dataset 2)

Model	Dataset 1	Dataset 2
AB	94.8	90.9
HARD	90.3	89.8
SOFT	95.7	**92.4**
LATENT	**95.8**	**92.4**

Table 3: Model Performance Comparison (MRR; %)

1,091. Linked English and Japanese titles are extracted, unless the Japanese title contains any other characters than Katakana, hyphen, or middle dot.

The language origin of titles were detected whether appropriate country names are included in the first sentence of Japanese articles. If they contain "ドイツの (of Germany)," "フランスの (of France)," "イタリアの (of Italy)," they are marked as German, French, and Italian origin, respectively. If the sentence contains any of Spain, Argentina, Mexico, Peru, or Chile plus "の"(of), it is marked as Spanish origin. If they contain any of America, England, Australia or Canada plus "の"(of), it is marked as English origin. The latter parts of Japanese/foreign titles starting from "," or "(" were removed. Japanese and foreign titles were split into chunks by middle dots and "_", respectively, and resulting chunks were aligned. Titles pairs with different numbers of chunks, or ones with foreign character length less than 3 were excluded. All accent marks were normalized (German "ß" was converted to "ss").

Implementation Details $P(c|s)$ of the class transliteration model was calculated by a character 3-gram language model with Witten-Bell discounting. Japanese Katakanas were all converted to Hepburn-style Roman characters, with minor changes so as to incorporate foreign pronunciations such as "wi / ウィ" and "we / ウェ." The hyphens "ー" were replaced by the previous vowels (e.g., "スパゲッティー" is converted to "supagettii.")

The maximum length of substitution pairs W described in Section 2 was set $W = 2$. The EM algorithm parameters $P(\alpha \rightarrow \beta|z)$ were initialized to the probability $P(\alpha \rightarrow \beta)$ of the alpha-beta model plus Gaussian noise, and π_k were uniformly initialized to $1/K$. Based on the preliminary results, we repeated EM iterations for 40 times.

5.2 Results

Language Class Detection We firstly show the precision of language detection using the class transliteration model $P(c|s)$ and Equation (3) (Table 5.2, 5.2). The overall precision is relatively lower than, e.g., Li et al. (2007), which is attributed to the fact that European names can be quite ambiguous (e.g., "Charles" can read "チャールズ ch$\bar{a}$ruzu" or "シャルル sharuru") The precision of Dataset 2 is even worse because it has more classes. We can also use the result of the latent class transliteration for clustering by regarding $k^* = \arg\max_k \gamma_{nk}$ as the class of the pair. The resulting cluster purity way was 0.74.

Transliteration Model Comparison We show the evaluation results of transliteration candidate retrieval task using each of $P_{AB}(t|s)$ (AB), $P_{\mathrm{hard}}(t|s)$ (HARD), $P_{\mathrm{soft}}(t|s)$ (SOFT), and $P_{\mathrm{latent}}(t|s)$ (LATENT) (Table 5.2). The number of latent classes was $K = 3$ for Dataset 1 and $K = 5$ for Dataset 2, which are the same as the numbers of language origins. LATENT shows comparable performance versus SOFT, although it can be higher depending on the value of K, as stated below. HARD, on the other hand, shows lower performance, which is mainly due to the low precision of class detection. The detection errors are alleviated in SOFT by considering the weighted sum of transliteration probabilities.

We also conducted the evaluation based on the top-1 accuracy of transliteration candidates. Because we found out that the tendency of the results is the same as MRR, we simply omitted the result in this paper.

The simplest model AB incorrectly reads "Felix / フェリックス," "Read / リード" as "フィリス *Firisu*" and "レアード *Reādo*." This may be because English pronunciation "x / ックス *kkusu*" and "ea /

イー *ī*” are influenced by other languages. SOFT and LATENT can find correct candidates for these pairs. Irregular pronunciation pairs such as “Caen / カーン *kān*” (French; misread “シャーン *shān*”) and “Laemmle / レムリ *Remuri*” (English; misread “リアム *Riamu*”) were misread by SOFT but not by LATENT. For more irregular cases such as “Hilda／イルダ *Iruda*”(English), it is difficult to find correct counterparts even by LATENT.

Finally, we investigated the effect of the number of latent classes K. The performance is higher when K is slightly smaller than the number of language origins in the dataset (e.g., $K = 4$ for Dataset 2) but the performance gets unstable for larger values of K due to the EM algorithm initial values.

6 Conclusion

In this paper, we proposed a latent class transliteration method which models source language origins as latent classes. The model parameters are learned from sets of transliterated words with different origins via the EM algorithm. The experimental result of Western person / proper name transliteration task shows that, even though the proposed model does not rely on explicit language origins, it achieves higher accuracy versus conventional methods using explicit language origins. Considering sources other than Western languages as well as targets other than Japanese is the future work.

References

Farooq Ahmad and Grzegorz Kondrak. 2005. Learning a spelling error model from search query logs. In *Proc. of EMNLP-2005*, pages 955–962.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling. In *Proc. ACL-2000*, pages 286–293.

Eric Brill, Gary Kacmarcik, and Chris Brockett. 2001. Automatically harvesting katakana-english term pairs from search engine query logs. In *Proc. NLPRS-2001*, pages 393–399.

Masato Hagiwara and Hisami Suzuki. 2009. Japanese query alteration based on semantic similarity. In *Proc. of NAACL-2009*, page 191.

Kevin Knight and Graehl Jonathan. 1998. Machine transliteration. *Computational Linguistics*, 24:599–612.

Haizhou Li, Khe Chai Sum, Jin-Shea Kuo, and Minghui Dong. 2007. Semantic transliteration of personal names. In *Proc. of ACL 2007*, pages 120–127.

Ariadna Font Llitjos and Alan W. Black. 2001. Knowledge of language origin improves pronunciation accuracy. In *Proc. of Eurospeech*, pages 1919–1922.