Cluster-specific Named Entity Transliteration

Fei Huang

School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213
fhuang@cs.cmu.edu

Abstract

Existing named entity (NE) transliteration approaches often exploit a general model to transliterate NEs, regardless of their origins. As a result, both a Chinese name and a French name (assuming it is already translated into Chinese) will be translated into English using the same model, which often leads to unsatisfactory performance. In this paper we propose a cluster-specific NE transliteration framework. We group name origins into a smaller number of clusters, then train transliteration and language models for each cluster under a statistical machine translation framework. Given a source NE, we first select appropriate models by classifying it into the most likely cluster, then we transliterate this NE with the corresponding models. We also propose a phrase-based name transliteration model, which effectively combines context information for transliteration. Our experiments showed substantial improvement on the transliteration accuracy over a state-of-the-art baseline system, significantly reducing the transliteration character error rate from 50.29% to 12.84%.

1 Introduction

Named Entity (NE) translation and transliteration are very important to many multilingual natural language processing tasks, such as machine translation, crosslingual information retrieval and question answering. Although some frequently occurring NEs can be reliably translated using information from existing bilingual dictionaries and parallel or monolingual corpora (Al-Onaizan and Knight, 2002; Huang and Vogel, 2002; Lee and Chang, 2003), less frequently occurring NEs, especially new names, still rely on machine transliteration to generate their translations.

NE machine transliteration generates a phonetically similar equivalent in the target language for a source NE, and transliteration patterns highly depend on the name's origin, e.g., the country or the language family this name is from. For example, when transliterating names [1] from Chinese into English, as shown in the following example, the same Chinese character "金" is transliterated into different English letters according to the origin of each person.

金人庆 --- **Jin** Renqing (China)
金大中 --- **Kim** Dae-jung (Korea)
马丁 路德 金 --- Martin Luther **King** (USA)
金丸信 --- **Kane**maru Shin (Japan)
何塞 华金 布伦纳 --- Jose Joa**quin** Brunner (Chile)

Several approaches have been proposed for name transliteration. (Knight and Graehl, 1997) proposed a generative transliteration model to transliterate foreign names in Japanese back to English using finite state transducers. (Stalls and Knight, 1998) expanded that model to Arabic-English transliteration. (Meng et al. 2001) developed an English-Chinese NE transliteration technique using pronunciation lexicon and phonetic mapping rules. (Virga and Khudanpur, 2003) applied statistical machine translation models to "translate" English names into Chinese characters for Mandarin spoken document retrieval. All these approaches exploit a general model for NE transliteration, where source names from different origins or language families are transliterated into the target language with the same rules or probability distributions, which fails to capture their different

[1] Assuming foreign names are already transliterated into Chinese.

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 435–442, Vancouver, October 2005.

transliteration patterns. Alternatively, (Qu and Grefenstette, 2004) applied language identification of name origins to select language-specific transliterations when back-transliterating Japanese names from English to Japanese. However, they only classified names into three origins: Chinese, Japanese and English, and they used the Unihan database to obtain the mapping between kenji characters and romanji representations.

Ideally, to explicitly model these transliteration differences we should construct a transliteration model and a language model for each origin. However, some origins lack enough name translation pairs for reliable model training. In this paper we propose a cluster-specific NE transliteration framework. Considering that several origins from the same language family may share similar transliteration patterns, we group these origins into one cluster, and build cluster-specific transliteration and language models.

Starting from a list of bilingual NE translation pairs with labeled origins, we group closely related origins into clusters according to their language and transliteration model perplexities. We train cluster-specific language and transliteration models with merged name translation pairs. Given a source name, we first select appropriate models by classifying it into the most likely cluster, then we transliterate the source name with the corresponding models under the statistical machine translation framework. This cluster-specific transliteration framework greatly improves the transliteration performance over a general transliteration model. Further more, we propose a phrase-based transliteration model, which effectively combines context information for name transliteration and achieves significant improvements over the traditional character-based transliteration model.

The rest of the paper is organized as following: in section 2 we introduce the NE clustering and classification schemes, and we discuss the phrase-based NE transliteration in section 3. Experiment settings and results are given in section 4, which is followed by our conclusion.

2 Name Clustering and Classification

Provided with a list of bilingual name translation pairs whose origins are already labeled, we want to find the origin clusters where closely related origins (countries sharing similar languages or cultural heritages) are grouped together.

We define the similarity measure between two clusters as their LM and TM perplexities. Let $S_i = \{(F_i, E_i)\}$ denote a set of name translation pairs from origin i, from which model θ_i is trained: $\theta_i = (P_{c(i)}, P_{e(i)}, P_{t(i)})$. Here $P_{c(i)}$ and $P_{e(i)}$ are N-gram character language models (LM) for source and target languages, and $P_{t(i)}$ is a character translation model trained based on IBM translation model 1 (Brown et.al. 1993). The distance between origin i and origin j can be symmetrically defined as:

$$d(i,j) = -\frac{1}{|S_i|}\log P(S_i \mid \theta_j) - \frac{1}{|S_j|}\log P(S_j \mid \theta_i),$$

where, assuming name pairs are generated independently,

$$P(S_i \mid \theta_j) \propto \sum_{t=1}^{|S_i|} \log[P_{c(j)}(F_i^t) P_{t(j)}(E_i^t \mid F_i^t) + P_{e(j)}(E_i^t) P_{t(j)}(F_i^t \mid E_i^t)]$$

We calculate the pair-wise distances among these origins, and cluster them with group-average agglomerative clustering. The distance between clusters C_i and C_j is defined as the average distance between all origin pairs in each cluster. This clustering algorithm initially sets each origin as a single cluster, then recursively merges the closest cluster pair into one cluster until an optimal number of clusters is formed.

Among all possible cluster configurations, we select the optimal cluster number based on the model perplexity. Given a held-out data set L, a list of name translation pairs from different origins, the probability of generating L from a cluster configuration Θ_ω is the product of generating each name pair from its most likely origin cluster:

$$P(L \mid \Theta_\omega) = \prod_{t=1}^{|L|} \max_{j \in \Theta_\omega} P(F^t, E^t \mid \theta_j) P(\theta_j)$$
$$= \prod_{t=1}^{|L|} \max_{j \in \Theta_\omega} P_{c(j)}(F^t) P_{e(j)}(E^t) P(\theta_j)$$

We calculate the language model perplexity:

$$pp(L, \Theta_\omega) = 2^{-\frac{1}{|L|}\log P(L \mid \Theta_\omega)} = P(L \mid \Theta_\omega)^{-1/|L|},$$

and select the model configuration with the smallest perplexity. We clustered 56K Chinese-English name translation pairs from 112 origins, and evaluate the perplexities of different models (number of

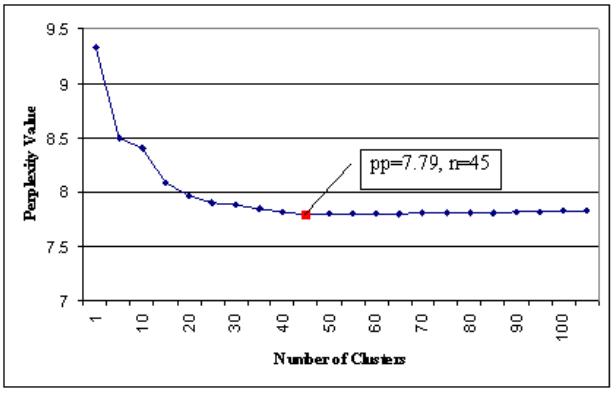

Figure 1. Perplexity value of LMs with different number of clusters

clusters) with regard to a held-out 3K name pairs. As shown in Figure 1, the perplexity curve reaches its minimum when $n = 45$. This indicates that the optimal cluster number is 45.

Table 1 lists some typical origin clusters. One may notice that countries speaking languages from the same family are often grouped together. These countries are either geographically adjacent or historically affiliated. For example, in the English cluster, the Netherlands (Dutch) seems an abnormality. In the clustering process it was first grouped with the South Africa, which was colonized by the Dutch and the English in the seventeenth century. This cluster was further grouped into the English-speaking cluster. Finally, some origins cannot be merged with any other clusters because they have very unique names and translation patterns, such as China and Japan, thus they are kept as single origin clusters.

For name transliteration task, given a source name *F* we want to classify it into the most likely cluster, so that the appropriate cluster-specific model can be selected for transliteration. Not knowing *F*'s translation *E*, we cannot apply the translation model and the target language model for name origin classification. Instead we train a Bayesian classifier based on N-gram source character language models, and assign the name to the cluster with the highest LM probability. Assuming a source name is composed of a sequence of source characters: $F = \{f_1, f_2, ..., f_l\}$. We want to find the cluster j^* such that

Arabic	Afghanistan, Algeria, Egypt, Iran, Iraq, Jordan, Kuwait, Pakistan, Palestine, Saudi Arabia, Sudan, Syria, Tunisia, Yemen, …
Spanish-Portuguese	Angola, Argentina, Bolivia, Brazil, Chile, Colombia, Cuba, Ecuador, Mexico, Peru, Portugal, Spain, Venezuela, …
English	Australia, Canada, Netherlands, New Zealand, South Africa, UK, USA, …
Russian	Belarus, Kazakhstan, Russia, Ukraine
East European	Bosnia and Herzegovina, Croatia, Yugoslavia
French (African)	Benin, Burkina Faso, Cameroon, Central African Republic, Congo, Gabon, Ivory Coast
German	Austria, Germany, Switzerland
French	Belgium, France, Haiti
Korean	North Korea, South Korea
Danish-Swedish	Denmark, Norway, Sweden
Single Clusters	China Japan Indonesia Israel ……

Table 1 Typical name clusters (n=45)

$$\begin{aligned} j^* &= \arg\max_j P(\theta_j \mid F) \\ &= \arg\max_j P(\theta_j) P(F \mid \theta_j) \\ &= \arg\max_j P(\theta_j) P_{c(j)}(F) \end{aligned} \quad (1)$$

where $P(\theta_j)$ is the prior probability of cluster *j*, estimated based on its distribution in all the training data, and $P_{c(j)}(F)$ is the probability of generating this source name based on cluster j's character language model.

3 Phrase-Based Name Transliteration

Statistical NE transliteration is similar to the statistical machine translation in that an NE translation pair can be considered as a parallel sentence pair, where "words" are characters in source and target languages. Due to the nature of name transliteration, decoding is mostly monotone.

NE transliteration process can be formalized as:

$$E^* = \text{argmax}_E\, P(E \mid F) = \text{argmax}_E\, P(F \mid E)P(E)$$

where E^* is the most likely transliteration for the source NE F, $P(F|E)$ is the transliteration model and $P(E)$ is the character-based target language model. We train a transliteration model and a language model for each cluster, using the name translation pairs from that cluster.

3.1 Transliteration Model

A transliteration model provides a conditional probability distribution of target candidates for a given source transliteration unit: a single character or a character sequence, i.e., "phrase". Given enough name translation pairs as training data, we can select appropriate source transliteration units, identify their target candidates from a character alignment path within each name pair, and estimate their transliteration probabilities based on their co-occurrence frequency.

A naive choice of source transliteration unit is a single character. However, single characters lack contextual information, and their combinations may generate too many unlikely candidates. Motivated by the success of phrase-based machine translation approaches (Wu 1997, Och 1999, Marcu and Wong 2002 and Vogel et. al., 2003), we select transliteration units which are long enough to capture contextual information while flexible enough to compose new names with other units. We discover such source transliteration phrases based on a character collocation likelihood ratio test (Manning and Schutze 1999). This test accepts or rejects a null hypothesis that the occurrence of one character f_1 is independent of the other, f_2, by calculating the likelihood ratio between the independent (H_0) and dependent (H_1) hypotheses:

$$\begin{aligned}\log\lambda &= \log\frac{L(H_0)}{L(H_1)} \\ &= \log L(c_{12}, c_1, p) + \log L(c_2 - c_{12}, N - c_1, p) \\ &- \log L(c_{12}, c_1, p_1) - \log L(c_2 - c_{12}, N - c_1, p_2)\end{aligned}$$

L is the likelihood of getting the observed character counts under each hypothesis. Assuming the character occurrence frequency follows a binomial distribution, $L(k,n,x) = \binom{n}{k} x^k (1-x)^{n-k}$, c_1, c_2, c_{12} are the frequencies of f_1, f_2 and $f_1 \wedge f_2$, and N is the total number of characters. p, p_1 and p_2 are defined as:

$$p = \frac{c_2}{N}, \quad p_1 = \frac{c_{12}}{c_2}, \quad p_2 = \frac{c_2 - c_{12}}{N - c_1}.$$

We calculate the likelihood ratio for any adjacent source character pairs, and select those pairs whose ratios are higher than a predefined threshold. Adjacent character bigrams with one character overlap can be recursively concatenated to form longer source transliteration phrases. All these phrases and single characters are combined to construct a cluster-specific phrase segmentation vocabulary list, T. For each name pair in that cluster, we

1. Segment the Chinese character sequence into a source transliteration phrase sequence based on maximum string matching using T;
2. Convert Chinese characters into their romanization form, *pinyin*, then align the pinyin with English letters via phonetic string matching, as described in (Huang et. al., 2003);
3. Identify the initial phrase alignment path based on the character alignment path;
4. Apply a beam search around the initial phrase alignment path, searching for the optimal alignment which minimizes the overall phrase alignment cost, defined as:

$$A^* = \arg\min_A \sum_{a_i \in A} D(f_i, e_{a_i}).$$

Here f_i is the ith source phrase in F, e_{a_i} is its target candidate under alignment A. Their alignment cost D is defined as the linear interpolation of the phonetic transliteration cost $\log P_{trl}$ and semantic translation cost $\log P_{trans}$:

$$D(f,e) = \lambda \log P_{trl}(e \mid f) + (1-\lambda)\log P_{trans}(e \mid f),$$

where P_{trl} is the product of the letter transliteration probabilities over aligned pinyin-English letter pairs, P_{trans} is the phrase translation probability calculated from word translation probabilities, where a "word" refers to a Chinese character or a English letter. More details about these costs are described in (Huang et. al., 2003). λ is a cluster-

specific interpolation weight, reflecting the relative contributions of the transliteration cost and the translation cost. For example, most Latin language names are often phonetically translated into Chinese, thus the transliteration cost is usually the dominant feature. However, Japanese names are often semantically translated when they contain characters borrowed from Chinese, therefore the translation cost is more important for the Japanese model (λ=0 in this case). We empirically select the interpolation weight for each cluster, based on their transliteration performance on held-out name pairs, and the combined model with optimal interpolation weights achieves the best overall performance.

We estimate the phrase transliteration probability according to their normalized alignment frequencies. We also include frequent sub-name translations (first, middle and last names) in the transliteration dictionary. Table 2 shows some typical transliteration units (characters or phrases) from three clusters. They are mostly names or sub-names capturing cluster-specific transliteration patterns. It also illustrates that in different clusters the same character has different transliteration candidates with different probabilities, which justifies the cluster-specific transliteration modeling.

Arabic	穆罕默德 mohamed 阿卜杜勒 abdul 艾哈迈德 ahmed
	尤: yo (0.27) y(0.19) you(0.14)…
English	约翰 john 威廉 william 彼得 peter
	尤: u(0.25) you(0.38) joo(0.16)…
Russian	弗拉基米尔 vladimir 伊万诺夫 ivanov –耶维奇 -yevich
	尤： yu(0.49) y(0.08) iu(0.07)…

Table 2. Transliteration units examples from three name clusters.

3.2 Language model and decoding

For each cluster we train a target character language model from target NEs. We use the N-gram models with standard smoothing techniques.

During monotone decoding, a source NE is segmented into a sequence of transliteration units, and each source unit is associated with a set of target candidate translations with corresponding probabilities. A transliteration lattice is constructed to generate all transliteration hypotheses, among which the one with the minimum transliteration and language model costs is selected as the final hypothesis.

4 Experiment Results

We selected 62K Chinese-English person name translation pairs for experiments. These origin-labeled NE translation pairs are from the name entity translation lists provided by the LDC [2] (including the who'swho (china) and who'swho (international) lists), and devided into three parts: system training (90%), development (5%) and testing (5%). In the development and test data, names from each cluster followed the same distribution as in the training data.

4.1 NE Classification Evaluation

We evaluated the source name classification accuracy, because classification errors will lead to incorrect model selection, and result in bad transliteration performance in the next step. We trained 45 cluster-specific N-gram source character language models, and classified each source name into the most likely cluster according to formula 1. We evaluated the classification accuracy on a held-out test set with 3K NE pairs. We also experimented with different *N* values. Table 3 shows the classification accuracy, where the 3-gram model achieves the highest classification accuracy. A detailed analysis indicates that some classification errors are due to the inherent uncertainty of some names, e. g, "骆家辉 (Gary Locke)", a Chinese American, was classified as a Chinese name based on its source characters while his origin was labeled as USA.

N=2	N=3	N=4	N=5	N=6	N=7
83.62	**84.88**	84.00	84.04	83.94	83.94

Table 3. Source name origin classification accuracies

[2] http://www.ldc.upenn.edu

4.2 NE Transliteration Evaluation

We first evaluated transliteration results for each cluster, then evaluated the overall results on the whole test set, where a name was transliterated using the cluster-specific model in which it was classified. The evaluation metrics are:

- Top1 accuracy (**Top1**), the percentage that the top1 hypothesis is correct, i.e., the same as the reference translation;
- Top 5 accuracy (**Top5**), the percentage that the reference translation appears in the generated top 5 hypotheses;
- Character error rate (**CER**), the percentage of incorrect characters (inserted, deleted and substituted English letters) when the top 1 hypothesis is aligned to the reference translation.

Our baseline system was a character-based general transliteration model, where 56K NE pairs from all clusters were merged to train a general transliteration model and a language model (**CharGen**). We compare it with a character-based cluster-specific model (**CharCls**) and a phrase-based cluster-specific model (**PhraCls**). The CERs of several typical clusters are shown in Table 4.

Because more than half of the training name pairs are from Latin language clusters, the general transliteration and language models adopted the Latin name transliteration patterns. As a result, it obtained reasonable performance (20-30% CERs) on Latin language names, such as Spanish, English and French names, but strikingly high (over 70%) CERs on oriental language names such as Chinese and Japanese names, even though the Chinese cluster has the most training data.

When applying the character-based cluster-specific models, transliteration CERs consistently decreased for all clusters (ranging from 6.13% relative reduction for the English cluster to 97% for the Chinese cluster). As expected, the oriental language names obtained the most significant error reduction because the cluster-specific models were able to represent their unique transliteration patterns. When we applied the phrased-based transliteration models, CERs were further reduced by 23% ~ 51% for most clusters, because the context information were encapsulated in the transliteration phrases. An exception was the Chinese cluster, where names were often translated according to the pinyin of single characters, thus phrase-based transliteration slightly decreased the performance.

The transliteration performance of different clusters varied a lot. The Chinese cluster achieved 96.09% top 1 accuracy and 1.69% CER with the character-based model, and other clusters had CERs ranging from 7% to 30%. This was partly because of the lack of training data (e.g, for the Japanese cluster), and partly because of unique transliteration patterns of different languages. We try to measure this difference using the average number of translations per source phrase (**AvgTrans**), as shown in Table 4. This feature reflected the transliteration pattern regularity, and seemed linearly correlated with the CERs. For example, compared with the English cluster, Russian names have more regular translation patterns, and its CER is only 1/3 of the English cluster, even with only half size of training data.

In Table 5 we compared translation examples from the baseline system (**CharGen**), the phrase-based cluster-specific system (**PhraCls**) and a online machine translation system, the **BabelFish**[3]. The **CharGen** system transliterated every name in the Latin romanization way, regardless of each name's original language. The **BabelFish** system inappropriately translated source characters based on their semantic meanings, and the results were difficult to understand. The **PhraCls** model captured cluster-specific contextual information, and achieved the best results.

We evaluated three models' performances on all the test data, and showed the result in Table 6. The **CharGen** model performed rather poorly transliterating oriental names, and the overall CER was around 50%. This result was comparable to other state-of-the-art statistical name transliteration systems (Virga and Khudanpur, 2003). The **CharCls** model significantly improved the top1 and top 5 transliteration accuracies from 3.78% to 51.08%, and from 5.84% to 56.50%, respectively. Consistently, the CER was also reduced from 50.29% to 14.00%. Phrase-based transliteration further increased the top 1 accuracy by 9.3%, top 5 accuracy by 10.7%, and reduced the CER by 8%, relatively. All these improvements were statistically significant.

[3] http://babelfish.altavista.com/

Cluster	Training data size	CharGen (CER)	CharCls (CER)	PhraCls (CER)	AvgTrans
Arabic	8336	22.88	18.93	14.47	4.58
Chinese	27093	76.45	1.69	1.71	3.43
English	8778	31.12	29.21	17.27	5.02
French	2328	27.66	18.81	9.07	3.51
Japanese	2161	86.94	38.65	29.60	7.57
Russian	4407	29.17	9.62	6.55	3.64
Spanish	8267	18.87	15.99	10.33	3.61

Table 4. Cluster-specific transliteration comparison

Cluster	Source	Reference	CharGen	PhraCls	BabelFish
Arabic	纳吉 萨布里 艾哈迈德	Nagui Sabri Ahmed	Naji Saburi Ahamed	Naji Sabri Ahmed	In natrium 吉 萨 cloth Aihamaide
Chinese	范志伦	Fan Zhilun	Van Tylen	Fan zhilun	Fan Zhilun
English	罗伯特 斯特德沃德	Robert Steadward	Robert Stdwad	Robert Sterdeward	Robert Stead Warder
French	让-吕克 科雷捷	Jean-luc Cretier	Jean-luk Crete	Jean-luc Cretier	Let - Lu Keke lei Jie
Japanese	小林隆治	Kobayashi Ryoji	Felinonge	Kobayashi Takaji	Xiaolin pros- perous gov- erns
Russian	弗拉基米尔 萨姆索诺夫	Vladimir Samsonov	Frakimir Samsonof	Vladimir Samsonov	弗拉基 mil sum rope Knoff
Spanish	鲁道夫 卡多索	Rodolfo Cardoso	Rudouf Cardoso	Rodolfo Cadozo	Rudolph card multi- ropes

Table 5. Transliteration examples from some typical clusters

Model	Top1 (%)	Top5 (%)	CER (%)
CharGen	3.78±0.69	5.84±0.88	50.29±1.21
CharCls	51.08±0.84	56.50±0.87	14.00±0.34
PhraCls	56.00±0.84	62.66±0.91	12.84±0.41

Table 6 Transliteration result comparison

5 Conclusion

We have proposed a cluster-specific NE transliteration framework. This framework effectively modeled the transliteration differences of source names from different origins, and has demonstrated substantial improvement over the baseline general model. Additionally, phrase-based transliteration further improved the transliteration performance by a significant margin.

References

Y. Al-Onaizan and K. Knight. 2002. Translating named entities using monolingual and bilingual resources. *In Proceedings of the ACL-2002*, pp400-408, Philadelphia, PA, July, 2002.

F. Huang and S. Vogel. 2002. Improved Named Entity Translation and Bilingual Named Entity Extraction, *Proceedings of the ICMI-2002*. Pittsburgh, PA, October 2002

F. Huang, S. Vogel and A. Waibel. 2003. Automatic Extraction of Named Entity Translingual Equivalence Based on Multi-feature Cost Minimization. *Proceedings of the ACL-2003, Workshop on Multilingual and Mixed Language Named Entity Recognition*. Sapporo, Japan.

K. Knight and J. Graehl. 1997. Machine Transliteration. *Proceedings of the ACL-1997*. pp.128-135, Somerset, New Jersey.

C. J. Lee and J. S. Chang. 2003. Acquisition of English-Chinese Transliterated Word Pairs from Parallel-Aligned Texts using a Statistical Machine Transliteration Model. *HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*. pp96-103, Edmonton, Alberta, Canada.

C. D. Manning and H. Schütze. 1999. Foundations of Statistical Natural Language Processing. MIT Press. Boston MA.

H. Meng, W. K. Lo, B. Chen and K. Tang. 2001. Generating Phonetic Cognates to Handle Named Entities in English-Chinese Cross-Language Spoken Document Retrieval. *Proceedings of the ASRU-2001*, Trento, Italy, December.2001

D. Marcu and W. Wong. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. *Proceedings of EMNLP-2002*, Philadelphia, PA, 2002

F. J. Och, C. Tillmann, and H. Ney. Improved Alignment Models for Statistical Machine Translation. pp. 20-28; Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora; University of Maryland, College Park, MD, June 1999.

Y. Qu, and G. Grefenstette. Finding Ideographic Representations of Japanese Names Written in Latin Script via Language Identification and Corpus Validation. ACL 2004: 183-190

P. Virga and S. Khudanpur. 2003. Transliteration of Proper Names in Cross-Lingual Information Retrieval. *Proceedings of the ACL-2003 Workshop on Multi-lingual Named Entity Recognition* Japan. July 2003.

S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venogupal, B. Zhao and A. Waibel. The CMU Statistical Translation System, *Proceedings of MT Summit IX* New Orleans, LA, USA, September 2003

D. Wu. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. Computational Linguistics 23(3):377-404, September 1997.