$PR + RQ \approx PQ$: Transliteration Mining Using Bridge Language

Mitesh M. Khapra *
Indian Institute of Technology
Bombay,
Powai, Mumbai 400076
India
miteshk@cse.iitb.ac.in

Raghavendra Udupa
Microsoft Research India,
Bangalore,
India
raghavu@microsoft.com

A. Kumaran
Microsoft Research India,
Bangalore,
India
a.kumaran@microsoft.com

Pushpak Bhattacharyya
Indian Institute of Technology
Bombay,
Powai, Mumbai 400076
India
pb@cse.iitb.ac.in

Abstract

We address the problem of mining name transliterations from comparable corpora in languages P and Q in the following resource-poor scenario:

- Parallel names in PQ are not available for training.
- Parallel names in PR and RQ are available for training.

We propose a novel solution for the problem by computing a common geometric feature space for P, Q and R where name transliterations are mapped to similar vectors. We employ Canonical Correlation Analysis (CCA) to compute the common geometric feature space using only parallel names in PR and RQ and without requiring parallel names in PQ. We test our algorithm on data sets in several languages and show that it gives results comparable to the state-of-the-art transliteration mining algorithms that use parallel names in PQ for training.

Introduction

The importance of the problem of name translation in cross-language tasks such as Machine Translation (MT) and Cross-Language Information Retrieval (CLIR) is well recognized by the Human Language Technology community (Chen et al. 1998), (Virga and Khudanpur 2003). In Machine Translation, many of the out-of-vocabulary words are names and "name dropping" and mis-translation of names degrade the quality of the translated text (Hermjakob, Knight, and Iii 2008). In CLIR, names form a significant fraction of query terms and translating them correctly correlates highly with the retrieval performance (Mandl and Hacker 2005), (Udupa et al. 2009a). In Web Search, names are particularly important as they are both highly frequent[1] in queries and very helpful in understanding the intent of the query. Given the importance of the problem, name transliteration has been extensively studied in the context of both MT and CLIR (Knight and Graehl 1998), (AbdulJaleel and Larkey 2003), (Li et al. 2009), (Al-Onaizan and Knight 2002).

In this work, we are interested in Transliteration Mining as it is known to give significantly better results than Transliteration Generation in CLIR (Udupa et al. 2009a) and can be employed for Named Entity Recognition in resource-poor languages (Klementiev and Roth 2006) and Cross-Language Named Entity Retrieval (Sproat, Tao, and Zhai 2006). Most Transliteration Mining algorithms involve a training phase where a model is learnt over training data consisting of parallel names in the source and target languages. For many combinations of languages of the world, very little or no training data is available. In such resource-poor scenarios, most of the currently known Transliteration Mining algorithms will not work well as they are crucially dependent on the availability of parallel names in the source and target languages. Therefore, a Transliteration Mining algorithm that can work well even in the absence of parallel names in the source and target languages is desirable.

We propose a novel Transliteration Mining algorithm that learns a transliteration similarity model for the language pair PQ even when no parallel names are available in PQ. Our algorithm, however, expects training data in the form of parallel names in PR and RQ to be available where R is a bridge language. This is not a very uncommon scenario in real-life especially when R is a resource-rich language such as English. For instance, it is easy to obtain English-Greek and English-Tamil parallel names on the Web but very difficult to get Greek-Tamil parallel names in sufficient number.

Our solution can be summarized as follows:

- We compute a common geometric feature space for P, Q, and R using parallel names in PR and RQ by a novel application of Canonical Correlation Analysis (CCA)(Hardoon, Szedmák, and Shawe-Taylor 2004).
- We compute the transliteration similarity of names in PQ as a function of the Euclidean distance between the corresponding feature vectors.

We show that our algorithm, while using only parallel names in PR and RQ, gives results comparable to those of state-of-the-art Transliteration Mining algorithms that make use of parallel names in PQ for training. We report results on several comparable corpora in English-Tamil, English-

[*] This work was done by Mitesh during his internship at Microsoft Research India.

[1] About 40% of the terms in web search queries are proper nouns (*e.g.* Texas) and about 30% of the query terms are common nouns (*e.g.* pictures) (Barr, Jones, and Regelson 2008). Furthermore, about 71% of web search queries contain at least one named entity (*e.g.* Harry Potter) (Guo et al. 2009).

Hindi and English-Russian using Kannada as the bridge language.

The rest of the paper is organized as follows: We begin by discussing important prior research related to our work. Next we propose a method for measuring similarity of names across languages. We show that a common feature space for names in the source and target languages can be determined by employing Canonical Correlation Analysis on parallel names. We then show that the common feature space can be learnt using a bridge language when no parallel names in the source and target languages are available for training. Next we describe our experimental setup and discuss results of the experiments. Finally, we discuss some avenues for future work and conclude the paper.

Related Work

Transliteration Mining has received substantial attention in the recent past. A discriminative approach which uses character n-grams as features was proposed by (Klementiev and Roth 2006) and applied on small-sized temporally aligned comparable news corpora in English-Russian. They also combined time-series similarity with the discriminative classifier to improve the accuracy of their system as the discriminative classifier by itself gave relatively poor accuracy. However, time series are meaningful for only those names which occur frequently in the corpora whereas a large majority of names appear only a few times. Another discriminative approach using language-specific features for phonetic similarity and time-series was proposed by (Sproat, Tao, and Zhai 2006).

A generative model for Transliteration Mining based on extended HMM model was introduced by (Udupa et al. 2009b). They used the generative model as part of their MINT system for mining transliterations from large and unaligned comparable corpora. Further, they used the same model to mine transliterations of out-of-vocabulary query terms from the top results of the first pass retrieval of CLIR systems and obtained impressive improvement in retrieval performance (Udupa et al. 2009a).

A probabilistic model using the notion of "productions" was proposed by (Pasternack and Roth 2009) and shown to give significantly better results on some benchmarks than the time-series based approach of (Klementiev and Roth 2006). However, the model is computationally inefficient as it takes seven hours for matching 727 single word names in English against 47772 Russian words (Pasternack and Roth 2009).

All the methods discussed till now are supervised methods and require name translations in the source and target languages for learning the model. An unsupervised constraint-driven learning algorithm was proposed by (Chang et al. 2009). The method makes use of romanization tables and constraints and gives significantly better results than (Klementiev and Roth 2006). However, the method relies crucially on language-specific mapping constraints which are generally not available for most languages.

Our method differs from all the above methods. Firstly, it does not require parallel names in PQ and can leverage parallel names PR and RQ where R is a bridge language such as English. Thus, it can be applied to many more language pairs than any of these methods. Secondly, it is language-agnostic and does not require any language-specific knowledge. Thirdly, it is computationally very efficient as it takes less than two minutes for matching around 1000 words in the source language against 50000 words in the target language.

Transliteration Equivalence

At the heart of Transliteration Mining is the problem of *Transliteration Equivalence*: given string p in language P and string q in language Q, determine whether p and q are transliterations of each other. In other words, Transliteration Equivalence is the problem of measuring the similarity of names across languages. Once the similarity is computed, it can be appropriately thresholded to determine whether the names are transliterations or not.

In this section, we describe how Canonical Correlation Analysis can be employed to measure the similarity of names across languages P and Q when parallel names in PQ are available at training time. In the next section, we describe how Canonical Correlation Analysis can be employed to measure the similarity of names across languages P and Q using R as the bridge language when parallel names in PQ are not available at training time.

Measuring Similarity of Names Across Languages

Consider two names, *Marie* and *Mary*, in English written in the Latin script. A simple intuitive method for computing the similarity between the two names is to represent them as feature vectors and compute the similarity of the two feature vectors. The features, for example, can consist of character unigrams and bigrams:

$$\begin{aligned}\phi\left(\textit{Marie}\right) &= \{m, a, r, i, e, ma, ar, ri, ie\}\\ \phi\left(\textit{Mary}\right) &= \{m, a, r, y, ma, ar, ry\}.\end{aligned}$$

All names in English can thus be represented in a common feature space defined by character unigrams and bigrams in English and the similarity between two names, $name1$ (with feature vector ϕ_1) and $name2$ (with feature vector ϕ_2), in English can be determined as follows:

$$Sim\left(name1, name2\right) = e^{-||\phi_1-\phi_2||^2/2\epsilon^2} \tag{1}$$

We can use the same approach to determine the similarity between a name in English (e.g. *Mary*) and another in Hindi (e.g. मॅरी) if we can find a way of representing the two names by vectors in a common feature space. Once we map names in different languages/scripts to the same feature space, we can compute their similarity using Equation 1 and determine whether the names are transliterations by a simple thresholding of the similarity score.

Finding a common feature space for names in different languages/scripts is a non-trivial problem as the feature spaces of the two languages are disjoint as the scripts are totally different. We describe a principled method for finding a common feature space using CCA in the next sub-section. In its simplest form, CCA learns two linear transformations A and B which can be used to map the feature vectors in the source and target languages to a common feature space:

$$\phi \rightarrow A^T\phi = \phi_s \in R^d \tag{2}$$

$$\psi \rightarrow B^T\psi = \psi_s \in R^d \tag{3}$$

Learning a Common Feature Space for PQ using Parallel Names in PQ

Given a sample of multivariate data with two views, CCA finds a linear transformation for each view such that the correlation between the projections of the two views is maximized. Consider a sample $Z = \{(x_i, y_i)\}_{i=1}^{n}$ of multivariate data where $x_i \in R^{d_1}$ and $y_i \in R^{d_2}$ are two views of the object. In our case, (p_i, q_i) , $i = 1, \ldots, N$ is the set of parallel names in PQ and x_i (and resp. y_i) is the feature vector for the name p_i (and resp. q_i) in language P (and resp. Q).

Let $X = [x_1, \ldots, x_n]$ and $Y = [y_1, \ldots, y_n]$. Assume that X and Y are centered[2], *i.e.*, they have zero mean. Let a and b be two directions. We can project X onto the direction a to get $U = [u_1, \ldots, u_n]$ where $u_i = a^T x_i$. Similarly, we can project Y onto the direction b to get the projections $V = [v_1, \ldots, v_n]$ where $v_i = b^T y_i$. The aim of CCA is to find a pair of directions (a, b) such that the projections U and V are maximally correlated. This is achieved by solving the following optimization problem:

$$\begin{aligned} \rho &= max_{(a,b)} \frac{< Xa, Yb >}{||Xa||||Yb||} \\ &= max_{(a,b)} \frac{a^T XY^T b}{\sqrt{a^T XX^T a}\sqrt{b^T YY^T b}} \end{aligned} \quad (4)$$

The objective function of Equation 4 can be maximized by solving the following generalized eigenvalue problem (Hardoon, Szedmák, and Shawe-Taylor 2004):

$$XY^T \left(YY^T\right)^{-1} YX^T a = \lambda^2 XX^T a \quad (5)$$

$$\left(YY^T\right)^{-1} YX^T a = \lambda b \quad (6)$$

The subsequent basis vectors can be found by adding the orthogonality of bases constraint to the objective function. Although the number of basis vectors can be as high as $\min\{Rank(X), Rank(Y)\}$, in practice, only the first few basis vectors are used since the correlation of the projections is high for these vectors and small for the remaining vectors. In the remainder of this paper, we refer to the first $d > 0$ basis vectors as A and B.

Figure 1 pictorially depicts the operation of CCA.

Transliteration Equivalence using Bridge Language

In the previous section, we saw how to determine a common feature space for names in PQ and how to map the names to the common feature space. However, the solution to the problem of Transliteration Equivalence critically depended on the availability of parallel names in PQ. In this section, we show how to accomplish the same thing without using parallel names in PQ but using a bridge language R.

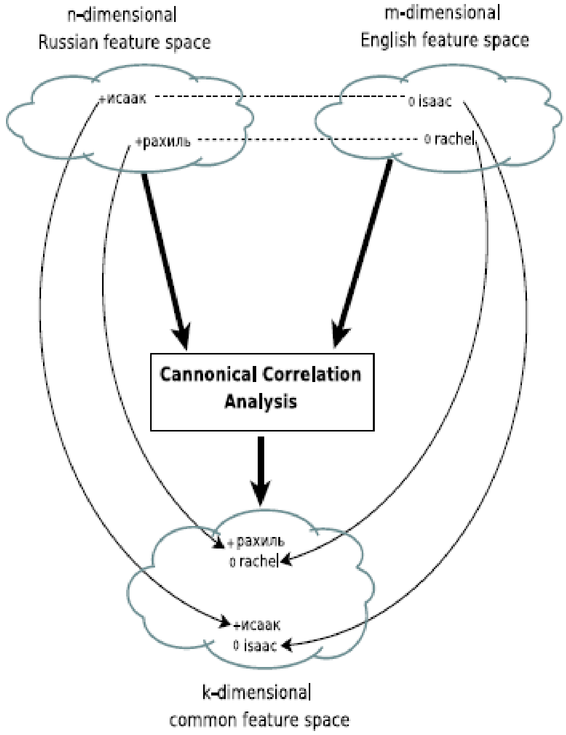

Figure 1: Canonical Correlation Analysis

Learning a Common Feature Space for PQ using R as the Bridge Language

We are given $m > 0$ parallel names $\{(p_i, r_i)\}_{i=1}^{m}$ in PR and $n > 0$ parallel names $\{(r_i, q_i)\}_{i=m+1}^{m+n}$ in RQ. Let

- $s_i \in R^{d_1}$ be the feature vector for the name p_i, $i = 1, \ldots, m$.
- $t_i \in R^{d_2}$ be the feature vector for the name r_i, $i = 1, \ldots, m$.
- $u_i \in R^{d_2}$ be the feature vector for the name r_i, $i = m + 1, \ldots, m + n$.
- $v_i \in R^{d_3}$ be the feature vector for the name q_i, $i = m + 1, \ldots, m + n$.
- $S = [s_1, \ldots, s_m] \in R^{d_1 \times m}$.
- $T = [t_1, \ldots, t_m] \in R^{d_2 \times m}$.
- $U = [u_{m+1}, \ldots, u_{m+n}] \in R^{d_2 \times n}$.
- $V = [v_{m+1}, \ldots, v_{m+n}] \in R^{d_3 \times n}$.

We treat each (p_i, r_i), $i = 1, \ldots, m$ as a single semantic object with two views $\left(s_i^T, \mathbf{0}\right)^T$ and t_i. One of the views, namely t_i, is due to the bridge language name r_i. The other view is a composite view consisting has two parts. The first part is due to the source language name p_i whereas the second part, due to the unknown target language transliteration of (r_i, p_i) is $\mathbf{0}$.

Similarly, we treat each (r_i, q_i), $i = m + 1, \ldots, m + n$ as a single semantic object with two views u_i and $\left(\mathbf{0}, v_i^T\right)^T$.

[2] If X and Y are not centered, they can be centered by subtracting the respective means

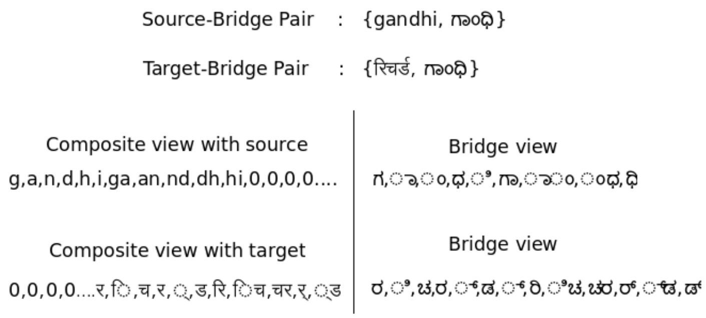
Source-Bridge Pair : {gandhi, ಗಾಂಧಿ}
Target-Bridge Pair : {रिचर्ड, ಗಾಂಧಿ}
Composite view with source
g,a,n,d,h,i,ga,an,nd,dh,hi,0,0,0,0....
Bridge view
ಗ,ಾ,ಂ,ಧ,ಿ,ಗಾ,ಾಂ,ಂಧ,ಧಿ
Composite view with target
0,0,0,0....र,ि,च,र,्,ड,रि,िच,चर,र्,्ड
Bridge view
ರ,ಿ,ಚ,ರ,್,ಡ,್,ರಿ,ಿಚ,ಚರ,ರ್,್ಡ,ಡ್

Figure 2: Views using Bridge Language with source language as English, target language as Hindi and bridge language as Kannada

One of the views, namely u_i, is due to the bridge language name r_i. As before, the other view has two parts. The first part of this view, due to the unknown source language transliteration of (r_i, q_i), is $\mathbf{0}$ whereas the second part is v_i and is due to the target language name q_i.

Thus, for every pair of parallel names in the training data, we can form two views:a first view x_i from the bridge language name r_i and a composite second view y_i from the counterpart of r_i as follows:

$$x_i = \left\{ \begin{array}{ll} t_i \in R^{d_2} & \text{if } 1 \leq i \leq m \\ u_i \in R^{d_2} & \text{if } m < i \leq m+n \end{array} \right\}.$$

$$y_i = \left\{ \begin{array}{ll} \left(s_i^T, \mathbf{0}\right)^T \in R^{d_1+d_3} & \text{if } 1 \leq i \leq m \\ \left(\mathbf{0}, v_i^T\right)^T \in R^{d_1+d_3} & \text{if } m < i \leq m+n \end{array} \right\}.$$

Figure 2 illustrates the construction of the two views of name pairs.

Let

$$\begin{aligned} X &= [x_1, \ldots, x_{m+n}] = \left[\begin{array}{cc} T & U \end{array}\right] \in R^{d_2 \times (m+n)}. \\ Y &= [y_1, \ldots, y_{m+n}] = \left[\begin{array}{cc} S & \mathbf{0} \\ \mathbf{0} & V \end{array}\right] \in R^{(d_1+d_3) \times (m+n)}. \end{aligned}$$

We can now use the machinery of CCA to find a common feature space for PQ. In fact, we can find a common feature space for PQR in which name transliterations in the three languages are mapped to similar vectors.

As in the previous section, we wish to find projection vectors $a \in R^{d_2}$ and $b \in R^{d_1+d_3}$ such that the projections $X^T a$ and $Y^T b$ are maximally correlated. Unlike before, the vector b can now be decomposed into two parts: $b = \left(b_1^T, b_2^T\right)^T$ where b_1 corresponds to the source language component of the composite view and b_2 corresponds to the target language component.

To find a, b_1 and b_2 we solve the following optimization problem stated in Equation 4 using the matrices X and Y formed as described earlier in this section:

$$\rho = max_{(a,b_1,b_2)} \frac{a^T T S^T b_1 + a^T U V^T b_2}{\sqrt{a^T (TT^T + UU^T) a} \sqrt{b_1^T S S^T b_1 + b_2^T V V^T b_2}} \quad (7)$$

It can be easily seen that the desired projection vectors are given by the solutions of the following generalized eigenvalue problem:

$$\begin{aligned} Ma &= \lambda^2 \left(TT^T + UU^T\right) a \\ ST^T \quad a &= \lambda S S^T b_1 \\ VU^T \quad a &= \lambda V V^T b_2 \end{aligned}$$

where $M = TS^T \left(SS^T\right)^{-1} ST^T + UV^T \left(VV^T\right)^{-1} VU^T$.

We refer to the first $d > 0$ triple of basis vectors as A, B_1 and B_2. We compute the similarity of a pair of names in PQ by projecting the source language name to the common feature space using B_1 and projecting the target language name using B_2 and computing the similarity between the two vectors using Equation 1.

Empirical Evaluation

We tested the effectiveness of our algorithm on data sets in several languages in several settings. In this section, we report the important results and analyze them. In the remainder of this section, we refer to our algorithm by the name BRIDGE-CCA.

Experimental Setup

As we wanted to compare the effectiveness of BRIDGE-CCA with a state-of-the-art Transliteration Mining algorithm, we used the same experimental setup as the one used by (Udupa et al. 2008) and (Udupa et al. 2009b). They use a two-stage approach called MINT for mining name transliterations from comparable corpora. In the first stage, for every document in the source language side, MINT finds a set of documents in the target language side with similar content using a KL-divergence based crosslanguage document similarity model. In the second stage, MINT extracts names from each source language document and matches them with candidates in the target language documents found in the first stage for this document. In our experiments, the first stage of MINT remained exactly the same. However, we used BRIDGE-CCA instead of the extended HMM model used by (Udupa et al. 2009b).

(Udupa et al. 2009b) proposed the following three data environments for evaluation:

IDEAL: Every article in the comparable corpora is aligned with exactly one similar article in the other language and the pairing of articles in the comparable corpora is known in advance.

NEAR-IDEAL: Every article in the comparable corpora is known to have exactly one conjugate article in the other language though the pairing itself is not known in advance.

REAL: For a given article in the source side of the comparable corpora, the existence of an equivalent article in the target side is not guaranteed.

We used the same comparable corpora described in (Udupa et al. 2009b) as summarized in Table 2. As we were concerned only with the mining stage of MINT, we used exactly the same article alignments produced by the first stage of MINT for the NEAR-IDEAL and REAL environments.

Test Bed	Environment	Mining Direction	Bridge Language	MRR		MRR of *BRIDGE-CCA* as % of MRR of MINT
				BRIDGE-CCA	MINT	
ET-ST	IDEAL	English-Tamil	Kannada	0.80	0.82	97.5%
EH-ST	IDEAL	English-Hindi	Kannada	0.82	0.93	88.1%
ER-ST	IDEAL	English-Russian	Kannada	0.77	0.79	97.4%
ET-ST	NEAR-IDEAL	English-Tamil	Kannada	0.68	0.84	80.9%
EH-ST	NEAR-IDEAL	English-Hindi	Kannada	0.78	0.86	90.6%
ET-LT	REAL	English-Tamil	Kannada	0.62	0.58	106.8%

Table 1: Performance of *BRIDGE-CCA* and MINT in the IDEAL, NEAR-IDEAL and REAL environments

Corpus	Language Pair	Data Environments	Article in Thousands		Words in Millions	
ET-S	English-Tamil	IDEAL, NEAR-IDEAL	2.90	2.90	0.42	0.32
EH-S	English-Hindi	IDEAL, NEAR-IDEAL	11.9	11.9	3.77	3.57
ER-S	English-Russian	IDEAL, NEAR-IDEAL	2.30	2.30	1.03	0.40
ET-L	English-Tamil	REAL	103.8	144.3	27.5	19.4

Table 2: Comparable Corpora

Training Data

Table 3 summarizes the training data used by our algorithm and the second stage of MINT (Udupa et al. 2009b). Note that we use no name pairs in the source and target languages for training our model. Further, the total number of name pairs used by us is substantially smaller than that used by (Udupa et al. 2009b). There were no common names in the source-bridge and bridge-target data sets.

Language Pair	Name Pairs	
	BRIDGE-CCA	MINT
English-Tamil	Eng-Kan (5K)+ Kan-Tam (5K)	∼16K
English-Hindi	Eng-Kan (5K)+ Kan-Hin (5K)	∼16K
English-Russian	Eng-Kan (5K)+ Kan-Rus (5K)	∼16K

Table 3: Training Data

Testbeds

We used the same testbeds as those used by (Udupa et al. 2009b). The testbeds for the IDEAL and NEAR-IDEAL environments are summarized in Table 4 and the testbed for REAL in Table 5.

Testbed	Comparable Corpora	Article Pairs	Distinct Name Pairs
ET-ST	ET-S	200	672
EH-ST	EH-S	200	373
ER-ST	ER-S	100	347

Table 4: Test Beds for IDEAL and NEAR-IDEAL

Testbed	Comparable Corpora	Article Pairs	Distinct Name Pairs
ET-LT	ET-L	100	228

Table 5: Test Bed for REAL

Features

We used character bigrams as features for all our experiments. For example, for representing the English word *Marie* the features used are { *ma, ar, ri, ie* }. Similarly, the features used for representing the Hindi word मॅरी are { मॅ, ॅर, री }. We also experimented with other features like unigrams, trigrams and combinations of unigrams, bigrams and trigrams. However, we found that the best performance was obtained using only bigrams and hence we used only bigrams for all our experiments.

Performance Measure

We used Mean Reciprocal Rank as the performance measure in our experiments:

$$MRR = \sum_{i=1}^{N} \frac{1}{r_i} \quad (8)$$

where r_i is the rank assigned by the mining algorithm to the correct transliteration of the ith source language name in the testbed. A MRR of 1 means that the mining algorithm always ranks the correct transliteration higher than any other candidate word. Therefore, a high MRR is an indication of the effectiveness of the mining algorithm.

Results

Table 1 compares the performance of our algorithm with that of MINT. We notice that even though BRIDGE-CCA uses substantially less training data than MINT and uses absolutely no parallel names in the source and target languages it is still able to compete with MINT which uses such data. For example, in the IDEAL environment, BRIDGE-CCA gave at least 88% of the performance of MINT and produced a very high MRR on all the testbeds. Similarly, in the NEAR-IDEAL environment, BRIDGE-CCA gave 80-90% of the performance of MINT. Finally, in the REAL environment, BRIDGE-CCA gave surprisingly better performance than MINT on the English-Tamil testbed.

BRIDGE-CCA does well even in terms of computational complexity as it took less than two minutes for mining all the transliterations in each of the data environments and on all the testbeds.

Apart from calibrating its performance in the bridge setting, we also wanted to test the strength of CCA as a classifier in general. For this we conducted experiments to check its performance when direct training data was available between PR. We found that in all the data environments the performance of CCA was within $\pm 2\%$ of the performance of the HMM based classifier used by MINT. Due to lack of space we have not reported the exact numbers here but its worth noting that BRIDGE-CCA gave 85-100% of the performance of DIRECT-CCA (*i.e.,* a CCA classifier trained using direct PR data). These results further highlight the strength of BRIDGE-CCA.

Future Work and Conclusions

We proposed a principled method for Transliteration Mining in a resource-poor scenario leveraging a bridge language. Our method does not need parallel names in the source and target languages and does not assume that the comparable corpora are temporally aligned. Further, it is language-agnostic as it does not make use of language-specific information. Experimental results show that the effectiveness of our method is comparable to that of a state-of-the-art algorithm that uses parallel names in the source and target languages for training.

As future work, we would like to see if a very small amount of direct data is available between PQ, can the model be suitably adjusted to include this data for training in addition to the data available between PR and RQ. We would also like to explore Transliteration Generation in the resource-poor scenario discussed in this work.

References

AbdulJaleel, N., and Larkey, L. S. 2003. Statistical transliteration for english-arabic cross language information retrieval. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*. ACM.

Al-Onaizan, Y., and Knight, K. 2002. Machine transliteration of names in arabic text. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*. Association for Computational Linguistics.

Barr, C.; Jones, R.; and Regelson, M. 2008. The linguistic structure of english web-search queries. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chang, M.-W.; Goldwasser, D.; Roth, D.; and Tu, Y. 2009. Unsupervised constraint driven learning for transliteration discovery. In *NAACL-HLT*.

Chen, H.-H.; Huang, S.-J.; Ding, Y.-W.; and Tsai, S.-C. 1998. Proper name translation in cross-language information retrieval. In *COLING-ACL*.

Guo, J.; Xu, G.; Cheng, X.; and Li, H. 2009. Named entity recognition in query. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 267–274. New York, NY, USA: ACM.

Hardoon, D. R.; Szedmák, S.; and Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation* 16(12):2639–2664.

Hermjakob, U.; Knight, K.; and Iii, H. D. 2008. Name translation in statistical machine translation - learning when to transliterate. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics.

Klementiev, A., and Roth, D. 2006. Named entity transliteration and discovery from multilingual comparable corpora. In *HLT-NAACL*.

Knight, K., and Graehl, J. 1998. Machine transliteration. *Computational Linguistics* 24(4):599–612.

Li, H.; Kumaran, A.; Pervouchine, V.; and Zhang, M. 2009. Report of news 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*. Association for Computational Linguistics.

Mandl, T., and Hacker, C. W. 2005. The effect of named entities on effectiveness in cross-language information retrieval evaluation. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*. ACM.

Pasternack, J., and Roth, D. 2009. Learning better transliterations. In *CIKM*, 177–186.

Sproat, R.; Tao, T.; and Zhai, C. 2006. Named entity transliteration with comparable corpora. In *ACL*.

Udupa, R.; Saravanan, K.; Kumaran, A.; and Jagarlamudi, J. 2008. Mining named entity transliteration equivalents from comparable corpora. In *CIKM*, 1423–1424.

Udupa, R.; Saravanan, K.; Bakalov, A.; and Bhole, A. 2009a. "they are out there, if you know where to look": Mining transliterations of oov query terms for cross-language information retrieval. In *ECIR*, 437–448.

Udupa, R.; Saravanan, K.; Kumaran, A.; and Jagarlamudi, J. 2009b. Mint: A method for effective and scalable mining of named entity transliterations from large comparable corpora. In *EACL*, 799–807.

Virga, P., and Khudanpur, S. 2003. Transliteration of proper names in cross-language applications. In *SIGIR*, 365–366.