Compositional Machine Transliteration

A KUMARAN
Microsoft Research India
MITESH M. KHAPRA[1] and PUSHPAK BHATTACHARYYA
Indian Institute of Technology Bombay

Machine Transliteration is an important problem in an increasingly multilingual world, as it plays a critical role in many downstream applications, such as machine translation or crosslingual information retrieval systems. In this paper, we propose compositional machine transliteration systems, where multiple transliteration components may be composed either to improve existing transliteration quality, or to enable transliteration functionality between languages even when no direct parallel names corpora exist between them. Specifically, we propose two distinct forms of composition – Serial and Parallel. Serial compositional system chains individual transliteration components, say, X $\rightarrow$ Y and Y $\rightarrow$ Z systems, to provide transliteration functionality, X $\rightarrow$ Z. In parallel composition evidence from multiple transliteration paths between X $\rightarrow$ Z are aggregated for improving the quality of a direct system. We demonstrate the functionality and performance benefits of the compositional methodology using a state of the art machine transliteration framework in English and a set of Indian languages, namely, Hindi, Marathi and Kannada. Finally, we underscore the utility and practicality of our compositional approach by showing that a CLIR system integrated with compositional transliteration systems performs consistently on par with and some time better than that integrated with a direct transliteration system.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms: Experimentation, Performance

Additional Key Words and Phrases: Machine Transliteration, Compositional Machine Transliteration, Transliterability, Resource Reusage, Multiple Evidence, Crosslingual Information Retrieval

[1]This work was done during the author's internship at Microsoft Research India.

This paper is being submitted to the Special Issue of ACM Transaction on Asian Language Information Processing on Information Retrieval for Indian Languages.

Author's address: A. Kumaran, Microsoft Research India, Bangalore, India.
Mitesh M. Khapra, Indian Institute of Technology-Bombay, Mumbai, India
Pushpak Bhattacharyya, Indian Institute of Technology-Bombay, Mumbai, India

1. INTRODUCTION

Machine Transliteration is an important problem in an increasingly multilingual world, for its critical role in many downstream application systems, such as Machine Translation (MT) and Crosslingual Information Retrieval (CLIR) systems. Proper names form an open set in any language, and they are shown to grow with the size of the corpora[2]. Names form a significant fraction of the user query terms, and handling them correctly correlates highly with the retrieval performance of the IR engine [Mandl and Womser-Hacker 2004]. In standard crosslingual evaluation datasets names are very prominent[3] and they affect the retrieval quality significantly[Mandl and Womser-Hacker 2005; Xu and Weischedel 2005; Udupa et al. 2009]. More importantly, the standard resources (such as, bilingual dictionaries) do not include name transliterations except for a small set of popular names, and keeping them updated continually is, in general, not an economically viable option. The statistical dictionaries, on the other hand, may not contain the transliterations as names are not frequent enough to provide sufficient statistical evidence during alignment[4]. Hence, the transliteration systems to rewrite the names in the target language are critically important in crosslingual scenarios. The importance of the transliteration problem is recognized well by the research community over the last couple of decades as evidenced by the increasing prominence for this topic in the research scope and publications of many Machine Translation, Information Retrieval, Natural Language Processing and Computational Linguistics conferences. The standard pair-wise transliteration systems are thoroughly researched and the approaches and performances are well published in research literature.

In this paper, we introduce the concept of Compositional Transliteration Systems as a composition of multiple transliteration systems to achieve transliteration functionality or to enhance the transliteration quality between a given pair of languages. We propose two distinct forms of composition – *serial* and *parallel*. In serial compositional systems, the transliteration systems are combined serially; that is, transliteration functionality between two languages X & Z may be created by combining transliteration engine X $\rightarrow$ Y and Y $\rightarrow$ Z. Such compositions may be useful for situations where no parallel data exists between two languages X & Z, but sufficient parallel names data may exist between X & Y, and Y & Z. Such partial availability of pair-wise data is common in many situations, where one central language dominates many languages of a country or a region. For example, there are 22 constitutionally recognized languages in India, but it is more likely that parallel names data might exist between Hindi and a foreign language, say, Russian, than between any other Indian language and Russian. In such situations, a transliteration system between Kannada, an Indian language, and Russian may be created by composing two transliteration modules, one between Kannada and Hindi, and the

[2]New names are introduced to the vocabulary of a language every day. On an average, 260 and 452 new names appeared on a daily basis in the XIE and AFE segments of the LDC English Gigaword corpora, respectively.

[3]Our own study of the topics from the 2004-07 CLEF [CLEF 2007] campaign revealed that 60% of the topics had at least one named entity, 39% had two or more, and 18% had three or more.

[4]Our analysis of *The Indian Express* news corpus over two years indicated that nearly 80% of the names occur less than 5 times in the entire corpus.

other between Hindi and Russian. Such compositions, if successful quality-wise, may alleviate the need for developing and maintaining parallel names corpora between many language pairs, and leverage the existing resources whenever possible, indicating a less resource intensive approach to develop transliteration functionality among a group of languages.

In parallel compositional systems, we explore combining transliteration evidence from multiple transliteration paths in parallel, in order to develop a good quality transliteration system between a pair of languages. While it is generally accepted that the transliteration quality of data-driven approaches grows with more data, typically the quality plateaus accruing only marginal benefit after certain size of the training corpora. In parallel compositional systems, we explore if transliteration quality between X & Z could be improved by leveraging evidences from multiple transliteration paths between X & Z. Such systems could be very useful when data is available between many different pairs among a set of n languages. Again, such situations naturally exist in many multicultural and multilingual societies, such as, India and the European Union. For example, parallel names data exists between many language pairs of the Indian subcontinent as most states enforce a 3-language policy, where all government records, such as census data, telephone directories, railway database, *etc.*, exist in English, Hindi and one of the regional languages. Similarly, many countries publish their parliamentary proceedings in multiple languages as mandated by legislative processes.

In our research we explore compositional transliteration functionality among a group of languages, and in this paper, our specific contributions are:

(1) Proposing the idea of compositionality of transliteration functionality, in two different methodologies: *serial and parallel*.
(2) Composing serially two transliteration systems – namely, X $\rightarrow$ Y and Y $\rightarrow$ Z – to provide a practical transliteration functionality between two languages X & Z with no direct parallel data between them.
(3) Improving the quality of an existing X $\rightarrow$ Z transliteration system through a parallel compositional methodology.
(4) Finally, demonstrating the effectiveness of different compositional transliteration systems – both serial and parallel – in an important downstream application domain of Crosslingual Information Retrieval.

We conduct a full set of experiments with a group of 4 languages of the Indian sub-continent, specifically, English, Hindi, Kannada and Marathi, between which parallel names corpora are available. We believe that such compositional transliteration functionality may be useful for many regions of the world, where common information access is necessary for political, social, cultural or economic reasons.

1.1 Related work

Current models for transliteration can be classified as grapheme-based, phoneme-based and hybrid models. Grapheme-based models, such as, Source Channel Model [Lee and Choi 1998], Maximum Entropy Model [Goto et al. 2003], Conditional Random Fields [Veeravalli et al. 2008] and Decision Trees [Kang and Choi 2000] treat transliteration as an orthographic process and try to map the source language

graphemes directly to the target language graphemes. Phoneme based models, such as, the ones based on Weighted Finite State Transducers [Knight and Graehl 1997] and extended Markov window [Jung et al. 2000] treat transliteration as a phonetic process rather than an orthographic process. Under such frameworks, transliteration is treated as a conversion from source grapheme to source phoneme followed by a conversion from source phoneme to target grapheme. Hybrid models either use a combination of a grapheme based model and a phoneme based model [Stalls and Knight 1998] or capture the correspondence between source graphemes and source phonemes to produce target language graphemes [Oh and Choi 2002].

Even though a wide range of algorithms have been developed for a variety of languages, there existed no consistent way of comparing these algorithms as the results were mostly reported on different datasets using different metrics. In this context, the shared task on Machine Transliteration in the recently concluded NEWS 2009 workshop [Li et al. 2009] was a successful attempt at calibrating different machine transliteration systems using common datasets and common metrics for a variety of language pairs. A study of various systems submitted to the workshop shows that grapheme based approaches performs better than or at par with phoneme based approaches, while requiring no specialized linguistic resources. In fact some of the best performing systems in the workshop were primarily grapheme based systems [Jiampojamarn et al. 2009; Jansche and Sproat 2009; Oh et al. 2009]. Further, combining any of the grapheme based engines with pre-processing modules like word-origin detection were shown to enhance the performance of the system [Oh and Choi 2002]. While previous research addressed combining evidence from multiple systems [Oh et al. 2009], to the best of our knowledge, ours is the first attempt at combining transliteration evidence from multiple languages.

However, a significant shortcoming of all the previous works was that none of them addressed the issue of performing transliteration in a resource scarce scenario, as there was an implicit assumption of availability of data between a pair of languages. In particular, we address a methodology to develop transliteration functionality between a pair of languages when no direct data exists between them. Some work on similar lines has been done in Machine Translation [Wu and Wang 2007] wherein an intermediate bridge language (say, Y) is used to fill the data void that exists between a given language pair (say, X and Z). In fact, recently it has been shown that the accuracy of a X$\rightarrow$Z Machine Translation system can be improved by using additional X $\rightarrow$ Y data provided Z and Y share some common vocabulary and cognates [Nakov and Ng 2009]. Similar work has also been done for transitive CLIR [Lehtokangas et al. 2008; Ballesteros 2000] where it was shown that employing a third language as an interlingua between the source and target languages, is a viable means of performing CLR between languages for which no bilingual dictionary is available. Specifically, Lehtokangas et al. [2008] automatically translated source language queries into a target language using an intermediate (or pivot) language and showed that such transitive translations were able to achieve 85-93% of the direct translation performance. Similarly, Gollins and Sanderson [Gollins and Sanderson 2001] proposed an approach called *triangulated transitive translation* which assumed the presence of two pivot languages for transitive CLIR. They showed that taking an intersection of the translations produced through two pivot

Table I. Language codes used for representing different languages

Language	Language Code
English	En
Hindi	Hi
Kannada	Ka
Marathi	Ma
Russian	Ru

languages can help to eliminate the noise introduced by each pivot language independently. The serial compositional approach described in this paper can be seen as an application of the transitive CLIR idea to the domain of machine transliteration. Similarly, the parallel compositional approach can be seen as a means of eliminating noise by taking multiple transliteration paths (as in the case of the *triangulated transitive translation* approach [Gollins and Sanderson 2001])

1.2 Organization

This paper is organized in the following manner. This section introduces the concept of compositional transliteration. This section also outlines the state of the art in transliteration systems research, and related work in machine translation scenarios. Section 2 outlines a language-independent orthography-based state of the art transliteration system that is used for all our experiments subsequently in this paper. Section 3 defines a measure that correlates well with the ease of transliteration between a given pair of languages. Section 4 introduces serial composition of transliteration systems and shows how a practical transliteration functionality may be developed between two languages. Section 5 introduces parallel composition of transliteration systems for combining evidence from multiple transliteration paths to improve the quality of the transliteration between a given pair of languages. Section 6 demonstrate effectiveness of such compositional systems in a typical usage scenario – Crosslingual Information Retrieval. Finally, Section 7 concludes the paper, outlining our future work.

1.3 Notation Used

Throughout the paper, we represent each language by its language code as described in Table I, and use the following convention to refer to a specific language or a transliteration system between a pair of languages: L_1-L_2 means a system for transliterating words from language L_1 to language L_2. For example, by *En-Hi* we mean a transliteration system from English to Hindi.

2. A GENERIC TRANSLITERATION SYSTEM

In this section, we outline the development of a language-neutral transliteration system that is to be used for all subsequent transliteration experiments.

2.1 A Generic Transliteration Engine between English and Indian Languages

First we set out to design a generic transliteration engine, so as to have a common system that can be used for establishing the baseline performance and the relative performance of various compositional transliteration alternatives. In addition we

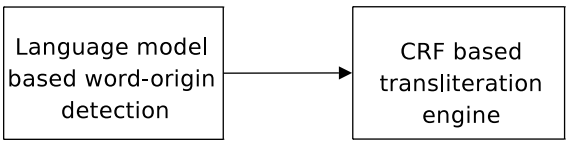

Fig. 1. Transliteration Engine Design

imposed a quality requirement that such a system work well across a wide variety of language pairs.

Systematic analysis of the various systems that participated in the NEWS 2009 shared task revealed that while the systems using phonetic information require additional linguistic resources, they perform only marginally better than purely orthographic systems. Further, amongst various machine learning techniques used for transliteration (using orthography or phonology), Conditional Random Fields based approach was the most popular among those participants in the first quartile. Hence, we decided to adopt a Conditional Random Fields based approach using purely orthographic features. In addition, since the Indian languages share many characteristics among them, such as distinct orthographic representation for different variations – aspirated or unaspirated, voiced or voiceless, etc. – of many consonants, we introduced a word origin detection module to identify specifically Indian origin names. Use of such classifier allowed us to train a specific CRF based transliteration engine for Indian origin names, and thus scoring a better quality transliteration. All other names are transliterated through an engine that is trained on non-Indian origin names.

We developed a generic Conditional Random Fields based transliteration engine, with a name origin detection module as a pre-processor (see Figure 1). The details of the subsystems are provided below.

2.1.1 *CRF-based Model for Transliteration.* Conditional Random Fields [Lafferty et al. 2001] are undirected graphical models used for labeling sequential data. Under this model, the conditional probability distribution of the target word given the source word is given by,

$$P(Y|X;\lambda) = \frac{1}{Z(X)} \cdot e^{\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(Y_{t-1}, Y_t, X, t)} \tag{1}$$

where,

$$\begin{aligned} X &= source\ word \\ Y &= target\ word \\ T &= length\ of\ source\ word \\ K &= number\ of\ features \\ \lambda_k &= feature\ weight \\ Z(X) &= normalization\ constant \end{aligned}$$

CRF++[5], an open source implementation of CRF was used for training and further transliterating the names. GIZA++[6] [Och and Ney 2003], a freely available implementation of the IBM alignment models [Brown et al. 1993] was used to get character level alignments for the name pairs in the parallel names training corpora. Under this alignment, each character in the source word is aligned to zero or more characters in the corresponding target word. The following features are then generated using this character-aligned data (here e_i and h_i form the i-th aligned pair of characters form the source word and target word respectively):

—h_i and e_j such that $i - 2 \leq j \leq i + 2$
—h_i and source character bigrams ($\{e_{i-1}, e_i\}$ or $\{e_i, e_{i+1}\}$)
—h_i and source character trigrams ($\{e_{i-2}, e_{i-1}, e_i\}$ or $\{e_{i-1}, e_i, e_{i+1}\}$ or $\{e_i, e_{i+1}, e_{i+2}\}$)
—h_i, h_{i-1} and e_j such that $i - 2 \leq j \leq i + 2$
—h_i, h_{i-1} and source character bigrams
—h_i, h_{i-1} and source character trigrams

The CRF model lends itself for fine-tuning to achieve optimal performance by experimenting with various configurations and yet applicable for a wide variety of language pairs. Further, this model may be trained only based on a training set of name pairs from the respective languages, without relying on any special linguistic tools or resources. While our experiments and analyses are confined to English and a set of Indian languages, it would be interesting to explore how it may scale for handling ideographic languages (such as, Chinese) or Semitic languages (such as, Arabic and Hebrew).

2.1.2 *Word Origin Detection.* Word origin detection is important for transliteration between English and Indian languages, specifically due to the difference in phonology between English and languages in the Indian subcontinent. While this is true in most transliteration systems, they play a crucial role in Indic names, as many variations for consonants typically exist in Indic language phonology.

To emphasize the importance of Word Origin Detection we consider the example of letter `d`. When `d` appears in a name of Western origin (e.g. `Daniel`, `Hudson`, `Alfred`) and is not followed by the letter `h`, it invariably gets transliterated as Hindi letter ड, whereas, if it appears in a name of Indic origin (e.g. `Devendra`, `Indore`, `Jharkhand`) then it is equally likely to be transliterated as द or ड. This shows that the decision is influenced by the origin of the word. Since the datasets (namely, Hindi, Kannada, Russian and Tamil) for the NEWS 2009 shared task consisted of a mix of Indic and Western names, it made sense to train separate models for words of Indic origin and words of Western origin.

For word origin detection, the words in the training data needed to be separated based on their origin. We first manually classified a random subset of the training set into of Indic origin names and Others. Two n-gram language models were built, for each of the already classified names of Indic origin and another for others. Each of the remaining names in the training corpora were split into a sequence of

[5]`http://crfpp.sourceforge.net/`
[6]`http://sourceforge.net/projects/giza/`

characters and the probability of such sequences using the two language models were constructed. Based on the computed probability, we classify all the name pairs in the training set as Indic names or others.

2.2 NEWS 2009 Transliteration Shared Task: Data & Systems

In the transliteration shared task conducted as a part of the ACL NEWS 2009 workshop [Li et al. 2009], 28 academic and industry groups from around the world participated in 8 diverse language pairs. The shared task published between 6K and 30K name pairs in various languages as training corpus, and the performances of systems on a common test corpora of about 1000 names in each language pair were published, highlighting the effect of various transliteration approaches on quality in different language pairs. For all our experiments in this section, we used only the training data published by the NEWS 2009 workshop (namely, approximately 6K name pairs in En-Ru, 8K name pairs in each of En-Ta and En-Ka, and 10K name pairs in En-Hi), and the test data for producing our results.

For word origin detection, 3K names were randomly chosen from the training corpus, and were manually annotated as Indian or Other. These 3K names were then divided into 4 non-overlapping folds. A 4-fold validation was performed using this data. In each case, we used 3 folds (*i.e.*, 2250 names) as training data for deriving language models and the remaining 4-th fold as test data. The average accuracy over the 4 folds was 97% *i.e.*, the test words were classified into Indic and Other origin names with an accuracy of approximately 97% . The above classifier was then again trained using the entire 3K names and was then applied on the entire data to yield reasonably well classified data that is used for training two distinct CRF-based modules for transliterating Indic and other names.

2.3 Transliteration Quality and Comparison with NEWS 2009 Participants

In this section we compare our experimental results on 4 language pairs (specifically, En-Hi, En-Ka, En-Ta and En-Ru) with that of the participating systems of the NEWS 2009 transliteration task. We used only the same training and test data that were released for NEWS 2009 Machine Transliteration Shared Task [Li et al. 2009], and hence the output were for *standard runs*, in NEWS 2009 parlance (that is, no extra data other than what was released for NEWS 2009 shared task, or no other linguistic tools or resources, were used). The top-10 transliteration candidates for each word were generated, and evaluated. The performance of our system is shown with the 3 standard measures as defined in [Li et al. 2009]: Specifically, the Word Accuracy in Top-1 (ACC-1), Fuzziness in Top-1 (F-score) and Mean Reciprocal Rank (MRR). As can be seen in Table II, our system was comparable to the best of the systems in the NEWS shared task, and would have been in the top quarter, in terms of ranking. We also want to highlight that the best system in NEWS 2009 [Jiampojamarn et al. 2009] used an online discriminative training sequence prediction algorithm using many-to-many alignments between the target and source. The Margin Infused Relaxed Algorithm (MIRA) [Crammer and Singer 2001] was used for learning the weights of the discriminative model. The second best system [Oh et al. 2009] in NEWS 2009 used a multi-engine approach wherein the outputs of multiple engines (Maximum Entropy Model, Conditional Random Fields and MIRA) were combined using different re-ranking functions.

Table II. Comparison of our System with the Best Systems of NEWS 2009

Language Pair	Our system			Best system in NEWS 2009			Rank of our system in NEWS 2009
	ACC-1	F-score	MRR	ACC-1	F-score	MRR	
En-Ru	0.604	0.927	0.693	0.613	0.928	0.696	3/13
En-Hi	0.417	0.877	0.546	0.498	0.890	0.603	7/21
En-Ta	0.420	0.898	0.549	0.474	0.910	0.608	4/14
En-Ka	0.354	0.869	0.476	0.398	0.880	0.526	5/14

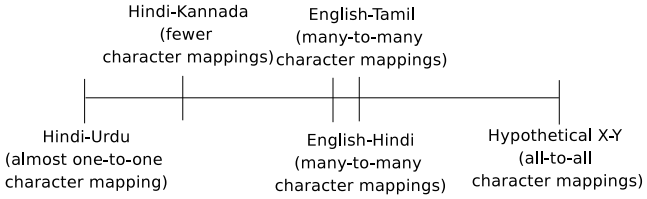

Fig. 2. The closeness of languages and transliterability

3. TRANSLITERABILITY AND TRANSLITERATION PERFORMANCE

In this section, we explore quantification of the ease of transliteration between a given language pair and using such knowledge for appropriate selection of language pairs for the composition of transliteration functionalities, and the selection of appropriate intermediate language for composition.

3.1 Language, Phonology, Orthography and Ease of Transliteration

In general, transliteration between a pair of languages is a non-trivial task, as the phonemic set of the two languages are rarely the same, and the mapping between phonemes and graphemes in respective languages are rarely one-to-one. However, many languages share a largely overlapping phoneme set (perhaps due to the geographic proximity or due to common evolution), and share many orthographic and/or phonological phenomenon. On one extreme, specific languages pairs have near-equal phonemes and an almost one-to-one mapping between their character sets, such as Hindi and Urdu [Malik et al. 2008], two languages from Indian subcontinent. Other language pairs, share similar, but unequal phoneme sets, but similar orthography possibly due to common evolution, such as Hindi and Kannada, two languages from the Indian sub-continent, with many phonological features borrowed from Sanskrit. This suggests that if we were to arrange language pairs on an axis according to the ease of transliterability between them then we would get a spectrum as shown in Figure 2. At one end of the spectrum would be language-pairs like Hindi-Urdu, and at the other end would be a hypothetical pair of languages where every character of one could map to every character of the other, with most language pairs somewhere in between the two extremes.

Below, we formulate a measure for transliterability that could correlate well with the transliteration performance of a generic system for a given language pair, which

in some sense would capture the ease of transliterability between them. First, we enumerate desirable qualities for such a measure:

(1) Rely purely on orthographic features of the languages only (and hence, easily calculated based on parallel names corpora)
(2) Capture and weigh the inherent ambiguity in transliteration at the character level. (*i.e.*, the average number of target – or source – characters that each source – or target – character can map to)
(3) Weigh the ambiguous transitions for a given character, according to the transition frequencies. Perhaps highly ambiguous mappings occur only rarely.

Based on the above, we propose a orthography based *Transliterability* measure that we call Weighted AVerage Entropy ($WAVE$), as given in Equation 2. Note that $WAVE$ will depend upon the n-gram that is being used as the unit of source and target language names, specifically, unigram, bigram or trigrams. Hence, we term the measures as $WAVE_1$, $WAVE_2$ or $WAVE_3$, depending on whether uni-, bi- or tri-grams were used for computing the measure.

$$WAVE_{n\text{-}gram} = \sum_{i \in alphabet} \Big(\frac{frequency(i)}{\sum_{j \in alphabet} frequency(j)} \cdot Entropy(i) \Big) \tag{2}$$

where,

$$\begin{aligned}
alphabet &= \text{Set of uni-, bi- or tri-grams} \\
Entropy(i) &= - \sum_{k \in Mappings(i)} P(k|j) \cdot log(P(k|j)) \\
i, j &= \text{Source Language Unit (uni-, bi- or tri-grams)} \\
k &= \text{Target Language Unit (uni-, bi- or tri-grams)} \\
Mappings(i) &= \text{Set of target language uni-, bi- or tri-grams that } i \text{ can map to}
\end{aligned}$$

To motivate the above proposed measure, we show in Table III, the source characters unigram frequencies computed based on the parallel names corpora outlined in Section 4.2, indicating that the unigram `a` is nearly 150 times more frequent than the unigram `x` in English names. Clearly, capturing the ambiguities of `a` will be more beneficial than capturing the ambiguities of `x`. The $frequency(i)$ term in Equation 2 captures this and ensures that the unigram `a` is weighed more than unigram `x`. In Table IV, some sample unigrams of the source language and the target unigrams that they map on to are shown; the numbers in brackets indicate the number of times a particular mapping was observed in the parallel names corpora detailed in Section 4.2. While both `c` and `p` have the same fanout of 2, the unigram `c` has higher entropy than the unigram `p` as the distribution of the fanout is much more dispersed than that of the unigram `c`. The $Entropy(i)$ term in Equation 2 captures this information and ensures that `c` is weighed more than `p`. Hence, we maintain that the measure captures the importance of handling specific characters in the source language and the inherent ambiguity in character mappings between the languages.

Table III. Character frequencies in English names

Source Character	Occurrence Frequency
a	18952
n	7161
q	236
x	137

Table IV. Characters: fanouts and ambiguities

Source Character	Mappings (Frequency)
c	स (200), क (200)
p	प (395), null (5)

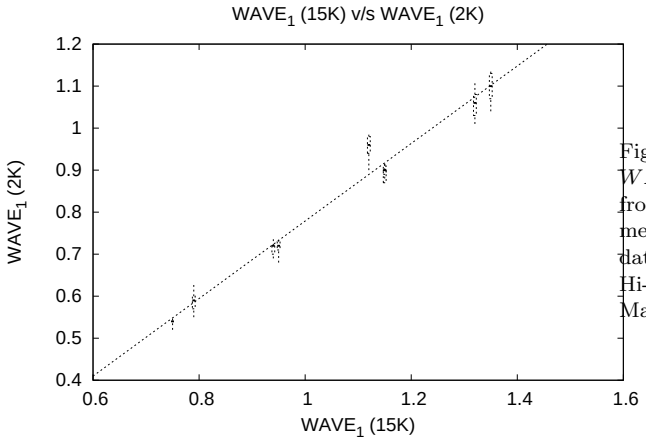

Fig. 3. Correlation between $WAVE_1$ measure calculated from 15K data and $WAVE_1$ measure calculated from 2K data for En-Hi, En-Ka, En-Ma, Hi-En, Ka-En, Ma-En, Hi-Ka, Ma-Ka

Next, we observed that $WAVE_{n\text{-}gram}$ can be computed fairly accurately, even with a small corpus. In Figure 3, we plot the $WAVE_1$ measures computed with 10 different samples of a 2K parallel names corpus (a randomly selected subset of the 15K corpus) and the entire 15K parallel names corpus, for various language pairs. The x-axis represents the $WAVE_1$ measure calculated from the 15K corpus and the y-axis represents a box and whisker plot based on the quartiles calculated from the 10 different samples of the 2K data. As can be seen the measures are highly correlated, suggesting that even a small corpus may be sufficient to capture the $WAVE_{n\text{-}gram}$ measures.

Finally, in Figure 4, we report the $WAVE_1$ measure, along with the maximum achieved quality of transliteration (for approximately 15K of training corpus) for the language pairs listed earlier. The x-axis plots the logarithm of the WAVE measure, and the y-axis the transliteration quality. We observe that as the WAVE measure increases the transliteration accuracy drops nearly linearly with logarithm of WAVE measure. In Figure 4, we present only the correlation between the *WAVE* measures and the transliteration quality achieved with a 15K training corpora. The two points in the top left corner in each of the plots represent transliteration between Hindi and Marathi languages that share the same orthography and have large one-to-one character mappings between them. Significantly (as shown in Figure 4),

we observe that different $WAVE_{n\text{-}gram}$ measures have similar effect on the transliteration quality, suggesting that even the uni-gram based WAVE measure captures the transliterability fairly accurately. Hence, for all subsequent experimentation, we used $WAVE_1$, as the uni-gram measure captures any correlation as accurately as other $WAVE_{n\text{-}gram}$ measures.

Based on the above observations, we term two languages with small $WAVE_1$ measure as more easily transliterable, and hence can be a candidate for either the first or the second component of any compositional transliteration systems involving one of these languages. Specific compositional transliteration experiments through an intermediate language and their performances are explored in the next section.

4. SERIAL COMPOSITIONAL TRANSLITERATION SYSTEMS

In this section, we address one of the configurations of the compositional transliteration systems – serial transliterations systems. Specifically, we explore the question *"Is it possible to develop a practical machine transliteration system between X and Z, by composing two intermediate $X \rightarrow Y$ and $Y \rightarrow Z$ machine transliteration systems?"* The utility of the compositional methodology is indicated by how close the performance of such a compositional transliteration system is to that of a direct transliteration system between X and Z.

4.1 Serial Compositional Methodology

It is a well known fact that transliteration is lossy, and hence it is expected that the composition of the two transliteration systems is only bound to have lower quality than that of each of the individual systems X $\rightarrow$ Y and Y $\rightarrow$ Z, as well as that of a direct system X $\rightarrow$ Z. We carry out a series of compositional experiments among a set of languages, to measure and quantify the expected drop in the accuracy of such compositional transliteration systems, with respect to the baseline direct system. We train two baseline CRF based transliteration systems (as outlined in Section 2), between the languages X and Y, and between the languages Y and Z, using appropriate parallel names corpora between them. For testing, each name in language X was provided as an input into X $\rightarrow$ Y transliteration system, and the top-10 candidate strings in language Y produced by the system were further given as an input into system Y $\rightarrow$ Z. The outputs of this system were merged and re-ranked by their probability scores. Finally, the top-10 of the merged outputs were output as the compositional system output.

To establish a baseline, the same CRF based transliteration system (outlined in Section 2) was trained with a 15K name pairs corpora between the languages X $\rightarrow$ Z. The performance of this system provides a baseline for a direct system between X & Z. The same test set used in the previous compositional systems testing was used for the baseline performance measurement in the direct system. As before, to avoid any bias, we made sure that there is no overlap between this test set and the training set for the direct system as well. The top-10 outputs were produced as the direct system output for comparison.

Additionally, we used the $WAVE_1$ measure, to effectively select the transition language between a given pair of languages. Given two languages X and Z, we chose a language that is easily transliterable to one of X or Z. The following experiments include both positive and negative examples for such transitions.

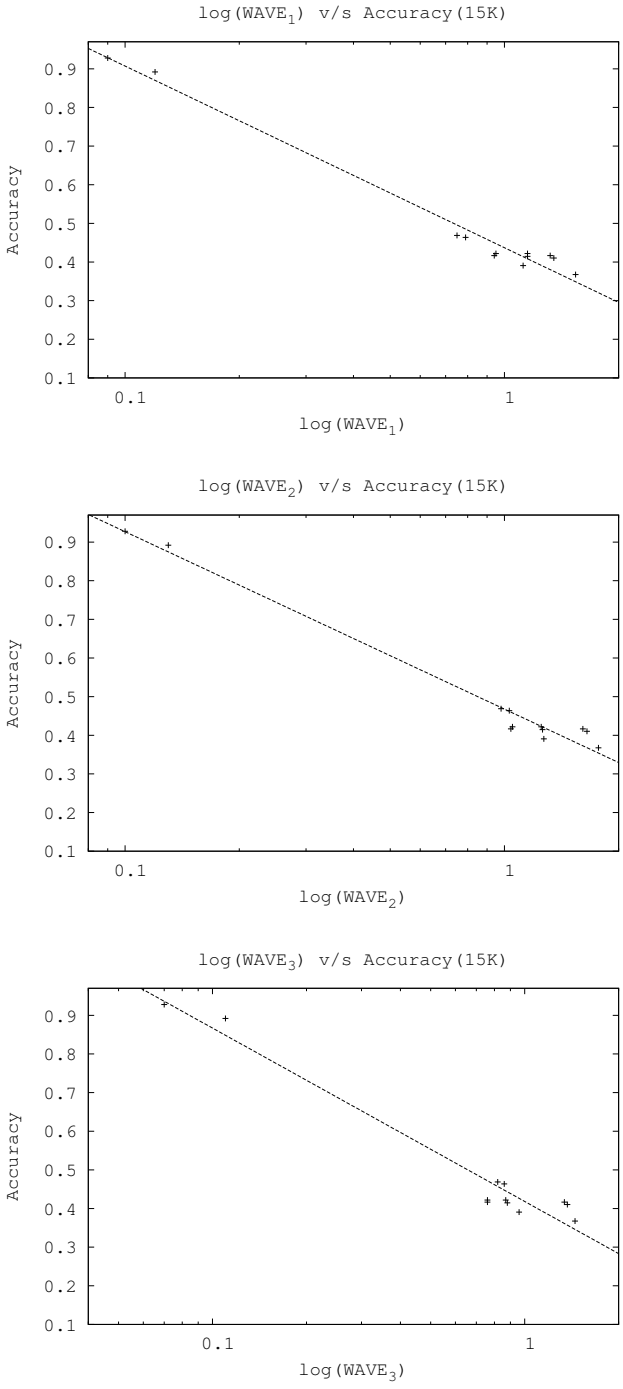

Fig. 4. **Correlation of WAVE with Transliteration Accuracy**

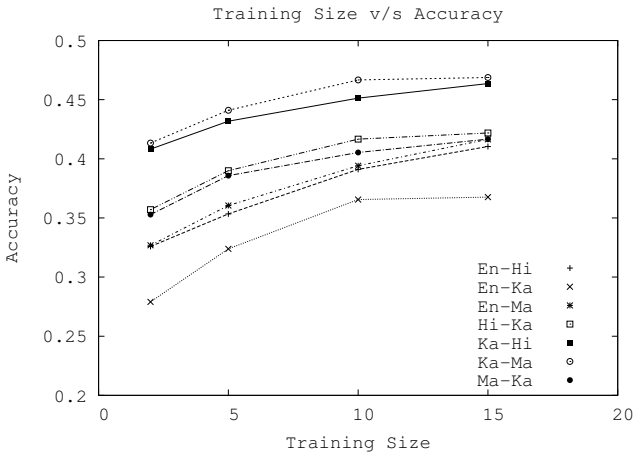

Fig. 5. Accuracy v/s size of training data

4.2 Data for Compositional Transliteration Systems

In this section, we detail the parallel names corpus that were used between English and a set of Indian languages for deriving correlation between $WAVE_{n\text{-}gram}$ metric and the transliteration performance between them. First, our transliteration experiments with the generic engine indicated that the quality of the transliteration increases continuously with data, but becomes asymptotic as the data size approaches 15K (see Figure 5) in all language pairs. Hence, we decided to use approximately 15K of parallel names corpora between English and the Indic languages (namely, Hindi, Kannada and Marathi), in all our subsequent experiments. While the NEWS 2009 training corpus ranged from 6K to 10K parallel names, we enhanced this training corpus in each language pair of interest (specifically, En-Hi, En-Ta and En-Ka) to 15K by adding more data of similar characteristics (such as, name origin, domain, length of the name strings, *etc.*), taken from the same source as the original NEWS 2009 data[7]. For other language pairs (such as, En-Ma) that were not part of the NEWS shared task, we created 15K parallel names corpora. We kept the test set in each of the languages the same as the standard NEWS 2009 test set. To avoid any bias, it was made sure that there is no overlap between the test set with the training sets of each of the X $\rightarrow$ Y and Y $\rightarrow$ Z systems.

4.3 Transliteration Performance of the Serial Compositional Systems

Table V details the experiments and the results of both the baseline direct systems and the compositional transliteration systems, in several sets of languages. All experiments list the three quality measures, namely, Accuracy (ACC-1), Mean Reciprocal Rank (MRR) and the Mean F-Score (F-Score) of both the direct and the compositional systems. For every experiment, a baseline system between the two languages (marked as X-Z) and the serial compositional system through an intermediate language (marked as X-Y-Z) are provided. Finally, the change in

[7]Since Microsoft Research India contributed the training and testing data to NEWS2009, we had access to larger parallel names corpus from which the NEWS 2009 data were derived.

Table V. Performance of Serial Compositional Transliteration Systems

Language Pair	**ACC-1**	**Δ ACC-1**	**MRR**	**Δ MRR**	F-score	**Δ F-score**
En-Ka (Baseline)	0.368		0.499		0.874	
En-Ma-Ka (Compositional)	0.347	-5.4%	0.460	-7.9%	0.862	-1.45%
En-Ma (Baseline)	0.416		0.547		0.879	
En-Hi-Ma (Compositional)	0.394	-5.3%	0.519	-5.3%	0.872	-0.74%
En-Ka (Baseline)	0.368		0.499		0.874	
En-Hi-Ka (Compositional)	0.334	-9.2%	0.440	-11.9%	0.852	-2.53%
Ka-En (Baseline)	0.391		0.492		0.878	
Ka-Hi-En (Compositional)	0.352	-9.9%	0.453	-7.8%	0.871	-1.14%
Ka-Hi (Baseline)	0.464		0.558		0.883	
Ka-En-Hi (Compositional)	0.267	-42.3%	0.366	-34.32%	0.819	-7.24%

the quality metric between the baseline direct and the compositional system, with respect to the quality of the baseline system, is also provided for every experiment.

Intuitively, one would expect that the errors of the first stage transliteration system (*i.e.*, X → Y) will propagate to the second stage (*i.e.*, Y→Z), leading to a considerable loss in the overall accuracy of the compositional system (*i.e.*, X → Y → Z). However, as we observe in Table V, the relative drop in the accuracy is less than 10%. For example, the baseline accuracy (ACC-1) of En-Ka baseline system is 0.368, where as the accuracy of the compositional En-Ma-Ka system is 0.347, a drop of a little more than 5%. The drop in mean reciprocal rank is under 12% and the drop in F-score is under 3%. The last system, namely the Ka-En-Hi, was chosen to illustrate the impact of a wrong choice of the intermediate language and is discussed specifically in Section 4.5.

4.4 Error Analysis in Serial Compositional Systems

The results shown in Table V contradict our basic intuition of massive degradation, and perhaps indicate that the two systems are not independent. To identify the reasons for the better than expected performance, we performed an error analysis of the output of each of the components of the serial compositional transliteration systems, to isolate the errors introduced at each stage.

Note that the first stage transliteration system (*i.e.*, X → Y) is expected to produce results according to the benchmarked quality (with respect to the generation of correct and incorrect transliterated strings in language Y). If the output of the stage 1 is correct, then we expect the stage 2 to produce results according to the benchmarked quality of the stage 2 (*i.e.*, Y → Z) system. On the other hand, when stage 1 produces incorrect transliterations, we expect stage 2 system to produce completely erroneous output, as input itself was incorrect. Contrary to our intuition, we find that many of the erroneous strings in language Y were actually getting corrected in Y → Z transliteration system, as shown by many examples in Table VI. For example, in the fourth example in Table VI, the Kannada string (`sumitomo`) gets incorrectly transliterated as सुमिटोमो (`sumitomo`) instead of सुमितोमो (`sumithomo`); however, for the second stage transliteration even this erroneous representation generates the correct English string (`sumitomo`). This interesting observation suggests that even though the input to the Y → Z system is an erroneous input in language Y from X → Y system, it still contains enough

Table VI. Examples of Errors in Ka→Hi→En Serial Transliteration System

Input Kannada string (Romanized)	**Erroneous Hindi by Ka → Hi (Stage 1) system**	**Correct Hindi (reference)**	**Correct English by Hi → En (Stage 2) system**
gularbhoj	गुलारभोज {gulaarbhoj}	गुलरभोज {gularbhoj}	gularbhoj
edana	एडाना {edaana}	एडना {edana}	edana
pakur	पकुर {pakur}	पाकुर {paakur}	pakur
sumitomo	सुमिटोमो {sumitomo}	सुमितोमो {sumithomo}	sumitomo

information for the Y → Z system to generate the correct output in language Z. However, note that this is possible only if the bridge language has richer orthographic inventory than the target language. For example, if we use a language such as Arabic, which drops all vowels, as the intermediate language, then we will not be able to recover the correct transliteration in the target language. In each of the successful bridge systems (that is, those with a relative performance drop of less than 10%), presented in Table V, the bridge language has, in general, richer orthographic inventory than the target language.

To isolate how many of such Stage 1 errors are getting corrected in the Stage 2, we performed an exhaustive error analysis in 5 different compositional transliteration systems. In each of the systems, we hand created a set of approximately 1,000 3-way parallel test names to calibrate the quality at every stage of the compositional X → Y and Y → Z transliteration systems. In this 3-way parallel set, for a given name in X, we created the correct equivalent names in languages Y and Z, so we could verify the correctness of the transliterations at each stage of the compositional transliteration system. The results are provided in the tables VII through XI, where the rows represent the performance of the stage 1 system, and the columns represent the performance of the stage 2 system. In each row, we segregated the correct and incorrect transliteration outputs from the X → Y system (in the rows) and verified for each of the input (correct or incorrect) whether the Y → Z produced correct output or not. Hence, in Table VII, for example, the X → Y system produced 41% correct transliterations (*i.e.*, 21.5% + 19.5%) and 59% incorrect transliterations (*i.e.*, 11.8% + 47.1%). This is in line with the expected quality of the X → Y system. The first row corresponds to the correct and incorrect transliteration by the Y → Z system, in line with the transliteration quality of the Y → Z system, as the inputs were correct strings in language Y. While we expected the second row to produce incorrect transliterations nearly for all inputs (as the input itself was an incorrect transliteration in language Y), we find upto 25% of the erroneous strings in language Y were getting transliterated correctly in language Z (for example, about 11.8% among the *wrong* 59% input strings were getting corrected in Table VII).

We see the same phenomenon in each of the tables VII through XI, indicating that some amount of information is captured even in the wrong transliterations in stage 1 to result in the correct transliteration output by the stage 2.

Table VII. Error Analysis for En→Hi→Ka

En→Hi→Ka		**Hi → Ka (Stage-2)**	
		Correct	**Error**
En→Hi (Stage-1)	**Correct (41%)**	21.5%	19.5%
	Error (59%)	**11.8%**	47.2%

Table VIII. Error Analysis for Ka→Hi→En

Ka→Hi→En		**Hi → En (Stage-2)**	
		Correct	**Error**
Ka→Hi (Stage-1)	**Correct (46%)**	21.9%	24.1%
	Error (54%)	**13.5%**	40.5%

Table IX. Error Analysis for En→Ma→Ka

En→Ma→Ka		**Ma → Ka (Stage-2)**	
		Correct	**Error**
En→Ma (Stage-1)	**Correct (41.6%)**	23%	18.6%
	Error (58.4%)	**11.8%**	46.6%

4.5 Impact of WAVE Measure on Transliteration Quality

In all these experiments (except the last Ka-En-Hi system) the intermediate language in the serial compositional transliteration system was chosen to be one that is easily transliterable from the source language or to the target language (*i.e.*, low $WAVE_1$ scores). Table XII reports the WAVE scores of the two stages of the compositional system as well as the the WAVE score of the direct system. For example, the first row of Table XII discusses the case when Hindi was used as the intermediate language for English to Kannada transliteration. The first stage of this compositional system was an English to Hindi transliteration system and the second stage was a Hindi to Kannada transliteration system. The $WAVE_1$ score of the direct system (*i.e.* English to Kannada) was 1.52 whereas the $WAVE_1$ scores for the first and second stages (*i.e.*, English to Hindi and Hindi to Kannada respectively) were 1.34 and 0.93 respectively.

We note in Table V that in the first 4 compositional systems, the $WAVE_1$ scores of the intermediate systems were generally smaller than that of the direct system, and the drop in accuracy of each of these compositional systems was under 10% when compared to the direct system[8]. The last row of Table XII shows that the $WAVE_1$ score of the direct system (0.78) was much less than the $WAVE_1$

[8]The only exception is the third system where the $WAVE_1$ score (1.34) of stage 1 (English-Hindi) was slightly greater than the $WAVE_1$ score (1.29) of the direct system (English-Marathi). However, this was compensated by the nearly zero $WAVE_1$ score of the second stage (Marathi-Hindi) of this compositional system.

Table X. Error Analysis for En→Hi→Ma

En→Hi→Ma		**Hi → Ma (Stage-2)**	
		Correct	**Error**
En→Hi (Stage-1)	**Correct (41.2%)**	37.2%	4%
	Error (58.8%)	**2%**	56.8%

Table XI. Error Analysis for Ka→En→Hi

Ka→En→Hi		**En → Hi (Stage-2)**	
		Correct	**Error**
Ka→En (Stage-1)	**Correct (39.1%)**	16.6%	22.5%
	Error (60.9%	**10%**	50.9%

Table XII. $WAVE_1$ scores for the different stages of the serial compositional systems

Language Pair	Intermediate Language	Stage-1 of Serial Compositional system	Stage-2 of Serial Compositional system	$WAVE_1$ for the direct system	$WAVE_1$ for Stage-1 of Serial Compositional system	$WAVE_1$ for Stage-2 of Serial Compositional system
En-Ka	Hi	En-Hi	Hi-Ka	1.52	1.34	0.93
En-Ka	Ma	En-Ma	Ma-Ka	1.52	1.29	0.90
En-Ma	Hi	En-Hi	Hi-Ma	1.29	1.34	0.06
Ka-En	Hi	Ka-Hi	Hi-En	1.11	0.78	0.92
Ka-Hi	En	Ka-En	En-Hi	0.78	1.11	1.34

scores of the intermediate systems (1.11 and 1.34). Correspondingly, Table V shows that in this case the drop in accuracy was much higher (42.3%). An empirical conclusion that we draw is that the constituent $WAVE_1$ measures, surrogates for transliterability, may suggest successful candidate pairs and may flag inappropriate candidate pairs, for compositional systems.

4.6 Effect of Vowels in the Transliteration

A closer error analysis revealed that vowels play a crucial role in the transliteration experiments as in nearly all the transliteration systems, approximately 60% of the errors were due to the incorrectly transliterated vowels. We thus performed some oracle experiments to quantify the impact of correct transliteration of vowels on overall transliteration quality. First, using a given X→Y transliteration system, we generated transliterations in language Y for about 1,000 names in language X. The resulting quality of transliteration (indicated as *ACC-1 without vowel Oracle* in Table XIII) was in line with the expected quality of the X→Y system. Next, we compared the output strings and the gold set, after ignoring all the vowel and combining *matras* from the generated transliteration strings in language Y and the gold reference set, presented as *ACC-1 with vowel Oracle* in Table XIII). Equiva-

Table XIII. Impact of vowels on accuracy

Language Pair	**ACC-1 (with vowel Oracle)**	**ACC-1 (without vowel Oracle)**	**Δ ACC-1**
En-Hi	0.748	0.410	+82.4%
En-Ka	0.642	0.368	+74.5%
En-Ma	0.754	0.416	+81.3%
Hi-En	0.721	0.422	+70.9%
Ka-En	0.711	0.415	+71.3%
Ma-En	0.650	0.422	+54.02%
Hi-Ka	0.742	0.464	+59.9%
Ka-Ma	0.764	0.469	+62.9%
Ma-Ka	0.647	0.417	+55.2%
Hi-Ma	0.939	0.928	+1.2%
Ma-Hi	0.909	0.892	+1.9%

lently, we can say, that the consonants are provided by the X→Y system, and the vowels are inserted by an oracle.

The results presented in Table XIII clearly indicate that substantial improvement in transliteration quality may be achieved by handling vowels correctly in the transliteration between English and Indian languages, and among Indian languages. This opens up a significant future research opportunity.

5. PARALLEL COMPOSITIONAL TRANSLITERATION SYSTEMS

In this section, we address the parallel compositional transliterations systems, specifically, combining transliteration evidence from multiple transliteration paths. Our objective here is to explore the question *"Is it possible to combine evidence from multiple transliteration paths to enhance the quality of a direct transliteration system between X and Z?"*. The usefulness of such a compositional system is indicated by how much *above* the performance of such a system is to that of a direct transliteration system between X and Z. Any improvement in transliteration quality may be very useful in going beyond the plateau for a given language pair.

5.1 Parallel Compositional Methodology

In this section, we explore if data is available between X and multiple languages, then is it possible to improve the accuracy of the X→Z system by capturing transliteration evidence from multiple languages. Specifically, we explore whether the information captured by a direct X→Z system may be enhanced with a serial X→Y→Z system, if we have data between all the languages. We evaluate this hypothesis by employing the following methodology, assuming that we have sufficient (∼15K, as detailed in Section 4.2) pair-wise parallel names corpora between X, Y & Z. First we train a X→Z system, using the direct parallel names corpora between X & Z. This system is called *Direct System*. Next, we build a serially composed transliteration system using the following two components: First, a X→Y transliteration system, using the 15K data available between X & Y, and, second a fuzzy transliteration system Y→Z that is trained using a training set that pairs the top-k outputs of the above trained X→Y system in language Y for a given string in language X, with the reference string in language Z corresponding to the string in language X. We

Table XIV. Performance of Parallel Compositional Transliteration Systems

Language Pair	ACC-1	Δ ACC-1	MRR	Δ MRR	F-score	Δ F-score
Hi-En (Direct)	0.422		0.539		0.884	
Hi-Ma-En (Fuzzy)	0.430		0.557		0.893	
Compositional	0.456	+8.1%	0.566	+4.9%	0.900	+1.8%
Ma-En (Direct)	0.415		0.534		0.880	
Ma-Hi-En (Fuzzy)	0.431		0.557		0.896	
Compositional	0.444	+7.2%	0.558	+4.7%	0.897	+2.0%
Ka-En (Direct)	0.391		0.492		0.878	
Ka-Hi-En (Fuzzy)	0.355		0.464		0.870	
Compositional	0.401	+2.6%	0.509	+3.5%	0.887	+1.0%
En-Ma (Direct)	0.416		0.547		0.879	
En-Hi-Ma (Fuzzy)	0.401		0.491		0.868	
Compositional	0.426	+2.2%	0.555	+1.31%	0.879	+0.03%

call this system as *Fuzzy System*, as it utilizes top-k (possibly incorrect) output in the intermediate language Y. We believe that even an incorrect output may contain sufficient information not captured in the direct system as evidenced by the error analysis in Section 4.4. We combine the evidence from these two systems – *direct* and *fuzzy* – for a given transliteration task between X and Z as follows: we merge the top-k outputs from the direct system, with the top-k outputs from the fuzzy system, using the following weighted average measure,

$$Score(T) = \lambda * Score_{direct}(T) + (1 - \lambda) * Score_{fuzzy}(T) \quad (3)$$

$$0 < \lambda < 1$$

and re-rank the results based on the above calculated scores. Note that the above formulation of combining the output of two systems is similar to that used by [Al-Onaizan and Knight 2001] for combining the output of a grapheme based system with a phoneme based system. A similar strategy was also used by [Zhou et al. 2008] to re-rank the candidate transliterations by taking a weighted sum of the score assigned by a transliteration engine and the normalized hit-count obtained for a candidate transliteration using a web search engine.

5.2 Results of Parallel Compositional Methodology

We employed the above strategy and tested parallel compositional methodology for combining transliteration for four language pairs and the quality of the results using the previously mentioned metrics – namely, Accuracy (ACC-1), Mean Reciprocal Rank (MRR) and Mean F-Score (F-Score) – are shown in Table XIV. In each of the experiments, the metrics for 4 systems are reported – the direct (line 1) and fuzzy (line 2) components of the parallel compositional systems, and the overall quality once combined (line 3). The quality of the direct system (line 1) provides the baseline for the corresponding parallel compositional transliteration system. The λ parameter is set to 0.4 for the first two systems and to 0.6 for the last two systems (as explained in Section 5.3).

It is surprising that there is an increase in the ACC-1, up to 8%, from the direct X→Z system, by combining evidence from fuzzy X→Y→Z system. Such improvement in transliteration quality suggests that combining evidence using parallel com-

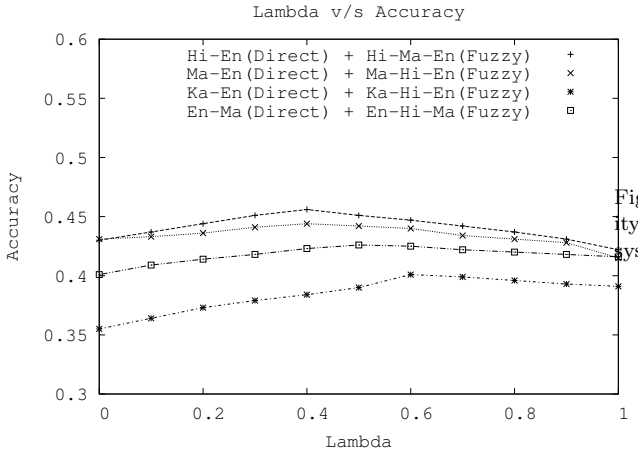

Fig. 6. Effect of λ on the quality of the parallel compositional system

position of transliteration engine may be productive, and may help go above the quality plateau achieved in direct systems.

5.3 Effect of varying λ

To study the effect of lambda on the quality of the composite system, we varied it from 0 to 1. A value of zero means only the fuzzy system's output was used and a value of 1 means only the direct system's output was used. Figure 6 shows a plot of the accuracies obtained for different values of λ. We observe that in each case, the best performance was obtained when λ was between 0.4 and 0.6. Further, the optimum value of λ depended on the quality of the direct system and the fuzzy system. Typically, if the quality of the direct system was better than the quality of the fuzzy system then the best results were obtained for $\lambda = 0.6$ (*i.e.,* when more weight was given to the output of the direct system). An example of this is the compositional system obtained by combining the Ka-En direct system with the Ka-Hi-En compositional system. On the other hand, if the quality of the fuzzy system was better than the quality of the direct system then the best results were obtained for $\lambda = 0.4$ (*i.e.,* when more weight was given to the output of the fuzzy system). An example of this is the compositional system obtained by combining the Hi-En direct system with the Hi-Ma-En compositional system.

6. EFFECTIVENESS OF COMPOSITIONAL TRANSLITERATION IN CLIR SYSTEM

In this section, we demonstrate the effectiveness of our compositional transliteration system on a downstream application, namely, a Crosslingual Information Retrieval system. We outline a standard state-of-the-art CLIR system for crosslingual document retrieval from a standard test collection. We specify the experimental set up and report the performance of the CLIR system integrated with compositional transliteration system, compared with a baseline integrated with a direct transliteration system.

6.1 CLIR System

We used a CLIR system that has been fielded for FIRE[9] 2008 shared task [FIRE 2008] for the CLIR experiments. Briefly, this CLIR system translates a given Hindi query (q_s) into English (q_t) using a probabilistic translation lexicon:

$$P(w_t|q_s) = \sum_{w_s} P(w_s|q_s)P(w_t|w_s) \tag{4}$$

where,

$$w_s = source\ word$$
$$w_t = target\ word$$
$$q_s = source\ query$$

Similarity of the translated query and a target document is measured using a Kullback-Leibler divergence based approach for scoring and ranking the documents, as follows:

$$Score(q_s, d_t) = \sum_{w_t, w_s} P(w_s|q_s)P(w_t|w_s)log(P(w_t|d_t)) \tag{5}$$

where,

$$d_t = target\ document$$

Details of our CLIR system are available in [Udupa et al. 2008]. This system was the best performing CLIR system between Hindi and English, with a MAP score of 0.4526, among a field of 8 participants in FIRE 2008.

6.2 Training and Test Document Sets for CLIR Experiments

The standard document collection used for FIRE 2008 shared task [FIRE 2008] was used for all our CLIR experiments. While the FIRE 2008 collection included documents in both English and multiple Indian languages, we used only Hindi to English portion of the FIRE 2008 CLIR experiments. The target document collection consists of 125,638 news articles in Indian English, from The Telegraph (Calcutta edition), gathered over a period of four years between 2004 and 2007. We used Hindi as the language of the query, specifically the topics 26-75 from the FIRE 2008 collection. All the three fields (title, description and narration) of the topics were used for the retrieval, as this setting would include all names in the query; note that names are the ones that are handled poorly by CLIR systems, and best helped by transliteration modules. Since the collection and topics are from previous years, their relevance judgements were also available as a reference for automatic evaluation. We used only the textual content of the documents for indexing and indexed only non-empty documents. The stop words are removed from the text while indexing and the words were stemmed using Porter Stemmer [Porter 1980].

[9]http://www.isical.ac.in/∼fire/

6.3 Linguistic Resources used for CLIR System

We used primarily the statistical dictionaries generated by training statistical word alignment models on an existing Hindi-English parallel corpora (∼100K parallel sentences between English and Hindi, consisting of about 70K words in English vocabulary and about 50K words in Hindi vocabulary), using the GIZA++ [Och and Ney 2003] tool. We used 5 iterations of IBM Model 1 and 5 iterations of HMM, retaining only the top 4 translations of every source word, along with their probability measures.

6.4 Integrating Machine Transliteration Systems in CLIR

As with any CLIR system that uses translation lexicon, we faced the problem of out-of-vocabulary (OOV) query terms that need to be transliterated, as they are typically proper names in the target language. First, for comparison, we used the above mentioned CLIR system with no transliteration engine, and measured the crosslingual retrieval performance. Clearly, the OOV terms would not be converted into target language, and hence contribute nothing to the retrieval performance. Second, we integrated a direct machine transliteration system between Hindi and English, which is expected to provide the correct transliterated strings in English, only in line with its transliteration performance. We report this performance as the baseline direct transliteration system performance. Third, we integrate, instead of a direct system, a set of serial compositional transliteration systems between Hindi and English, transitioning through different intermediate languages, namely Marathi and Kannada, and reported the CLIR performance for each of the compositional path. Finally, we integrate, a parallel compositional transliteration system, through Marathi as an intermediate language, where the results are combined with $\lambda = 0.4$, the best value as outlined in Section 5.3 for Hi-Ma-En system, and the CLIR performance measured and reported.

6.5 CLIR with Transliteration Systems Evaluation

The results of the above experiments are given in Table XV. The current focus of these experiments is to answer the question of *whether the compositional machine transliteration systems used to transliterate the OOV words in Hindi queries to English* (by stepping through an intermediate language – Marathi or Kannada) *performs at par with a direct transliteration system.*

We outline a series of experiments, in which the CLIR system integrated with different transliteration engines – both direct and compositional – perform on the standard FIRE 2008 data set. For these experiments, we used top-n ($n = 1,\ 5$ and 10) output of the integrated transliteration engine, and the results are reported separately. The following guide specifies the systems reported:

(1) *Baseline:* the baseline CLIR system with no transliteration engine integrated. This system performance is provided as the basis for quantifying the effect of transliteration on CLIR system performance.

(2) *D-Hi-En:* the baseline CLIR system, integrated with a direct machine transliteration system for transliterating OOV words between Hindi and English. This system provides a baseline for our compositional transliteration experiments.

Table XV. CLIR performance with differently configured transliteration systems

CLIR System	Description	Top-n	MAP	Δ MAP change from Baseline
Baseline	No Transliteration	-	0.4361	-
D-Hi-En	Direct	1	0.4546	+4.24%
S-Hi-Ka-En	Serial Compositional (via Kannada)	1	0.4617	+5.87%
S-Hi-Ma-En	Serial Compositional (via Marathi)	1	0.4664	+6.94%
P-Hi-Ma-En	Parallel Compositional (via Marathi)	1	0.4470	+2.49%
D-Hi-En	Direct	5	0.4549	+4.31% **
S-Hi-Ka-En	Serial Compositional (via Kannada)	5	0.4612	+5.75%
S-Hi-Ma-En	Serial Compositional (through Marathi)	5	0.4550	+4.33% **
P-Hi-Ma-En	Parallel Compositional (through Marathi)	5	0.4555	+4.44% **
D-Hi-En	Direct	10	0.4471	+2.52% **
S-Hi-Ka-En	Serial Compositional (through Kannada)	10	0.4621	+5.96%
S-Hi-Ma-En	Serial Compositional (through Marathi)	10	0.4543	+4.17% **
P-Hi-Ma-En	Parallel Compositional (through Marathi)	10	0.4470	+2.49% **

(3) *S-Hi-Ka-En:* the baseline CLIR system, integrated with a serial compositional machine transliteration system between Hindi and English transitioning through Kannada.

(4) *S-Hi-Ma-En:* the baseline CLIR system, integrated with a serial compositional machine transliteration system between Hindi and English transitioning through Marathi.

(5) *P-Hi-Ma-En:* the baseline CLIR system, integrated with a parallel compositional machine transliteration system between Hindi and English transitioning through Marathi.

As expected, enhancing the baseline CLIR system with a direct machine transliteration system (D-Hi-En) gives better results over a CLIR system with no transliteration functionality. Significantly, we observe that most of the compositional transliteration system perform on par or better than the direct system, at each output level. While the choice of the transition language and the compositional methodology has an influence on CLIR system between a given pair of languages, the on par results indicate that the compositional transliteration systems can be effectively employed in practical downstream applications. Two-tailed paired t-tests were performed to check whether the improvements in the MAP scores obtained by using the Direct, Serial and Parallel transliteration systems were statistically significant. The results marked with stars (**) in the 5th column of Table XV were found to be statistically significant with a confidence of 95% ($p = 0.05$). We observe that the improvements obtained by using the top-5 and top-10 transliterations were statistically significant. Also, the statistically significant results suggest that top-5 output produces the best improvement in the MAP scores, as expected in CLIR type applications.

Table XVI. Comparison of transliterations produced by different systems

Hindi query	D-Hi-En	S-Hi-Ka-En	S-Hi-Ma-En	P-Hi-Ma-En	English query
ग्रेग चैपल और सौरव गांगुली के बीच सशंक वइरामसधि	ghanguli (Incorrect)	ganguli (Incorrect)	ganguly (Correct)	ganguly (Correct)	Uneasy truce between Greg Chapell and Sourav Ganguly

A detailed analysis of the query translations produced by the above systems showed that in some cases the compositional system does produce a better transliteration thereby leading to a better MAP. As an illustration, consider the query containing the OOV name गांगुली {Ganguly} and the corresponding transliterations generated by the different systems as presented in Table XVI. The direct D-Hi-En system generated was unable to generate the correct transliteration in the top-5 results whereas the serial S-Hi-Ma-En system and the parallel & P-Hi-Ma-En system were able to produce the correct transliteration in the top-5 results thereby resulting in an improvement in MAP for this sample query. We also observe that as more number of top-n transliterations are added, the resulting MAP scores decreases slightly, perhaps due to the noise added by the wrong transliterations during query translation.

7. CONCLUSIONS & FUTURE RESEARCH DIRECTIONS

In this paper, we introduced the idea of compositional transliteration systems, where multiple transliteration components were composed, either to provide new transliteration functionality, or to enhance the existing transliteration quality, between a given pair of languages. Specifically, we proposed two distinct configurations – serial and parallel – for compositional systems. The serial compositional transliteration systems chained individual transliteration components in a serial manner, to enable creation of transliteration functionality for a given pair of languages with no parallel names corpora between them. Specifically, a transliteration system X $\rightarrow$ Z may be created, by composing X $\rightarrow$ Y and Y $\rightarrow$ Z transliteration components serially. Next, we explored the parallel compositional transliteration systems, which aggregated the transliteration evidence from multiple transliteration paths to improve the quality of a given transliteration system. Specifically, the quality of transliteration of X $\rightarrow$ Z system may be improved, by combining evidence from X $\rightarrow$ Y $\rightarrow$ Z systems.

We formulated a measure – $WAVE_{n\text{-}gram}$ – to measure the ease of transliteration (which we termed as *transliterability* between a given ordered language pair. We show how such a measure may help in designing serial compositional systems with minimal loss of quality. Further, such measure might help identifying appropriate languages between which parallel corpora needs to be developed, there by paving way for a less resource intensive approaches for providing transliteration functionality among a set of n languages.

To validate the utility of the compositional systems, we conducted a comprehensive set of experiments among English and 3 Indian languages, namely, Hindi, Marathi and Kannada. We conducted an extensive set of experiments to quantify

any change in the transliteration accuracy between a given pair of languages. First, we showed empirically that, quality-wise, the serial compositional systems do not degrade drastically, compared with baseline direct transliteration systems: The relative drop in accuracy of appropriately designed compositional systems is less than ∼10% of that of the corresponding direct systems, in general. Second, we performed an extensive stage-wise error analysis of the compositional systems, and identified that significant fraction of errors (∼25%) caused by the first stage transliteration system of the composition is getting corrected by the second stage transliteration system, providing an insight into the benefits of composition of transliteration components. Based on this insight, we designed parallel compositional transliteration systems, that combined evidence from a serial compositional system to a direct system, to improve the quality of the direct system. Empirically, we showed that there is a improvement of up to ∼8% in transliteration accuracy achieved by this methodology, over the direct transliteration systems. While the compositional methodology uses multiple datasets, each component may participate in many compositional systems thereby amortizing the development cost. In addition they may enable transliteration functionalities that may not be possible with the existing datasets, or improve transliteration quality above and beyond direct systems.

Finally, we showed that such compositional transliteration systems – both serial and parallel – may be used in practical situations effectively. We showed that a CLIR system working on the standard FIRE 2008 test collection between Hindi and English is helped by the integration of the compositional transliteration systems significantly, showing up to ∼8% improvement in MAP scores over the same CLIR system with no transliteration component. More significantly, these improvements are in-par with, and sometimes *better than*, the same CLIR system that had been integrated with a direct transliteration system between Hindi and English, thus establishing the practicality of using compositional transliteration systems.

7.1 Future Research Avenues

Transliteration is an important research area for downstream applications like CLIR or MT. However, there are many situations in which transliteration functionality needs to be developed among a set of languages, for political, social or economic reasons. Compositional systems provide a viable and practical solution in resource-scarce situations.

We plan to pursue the compositional transliteration functionality in several directions: First, we plan to expand the set of languages to explore the scalability of the compositional approaches for a diverse set of languages. Second, given a set of n languages, we would like to explore a principled way of selecting language pairs among the n languages, between which the transliteration corpus may be developed in order to balance the resource requirement and the transliteration accuracy. Finally, we would like to explore complex compositional approaches, involving more transliteration components arranged in more complex topologies.

Compositional systems may provide an effective way of enabling transliteration functionality among a group of languages, by reducing the need for developing resources in all combinations of languages, or using more effectively the available parallel corpora between languages. Ultimately, such approaches may help in reducing the digital divide that exist in many resource-poor parts of the world.

ACKNOWLEDGEMENTS

We thank the NEWS 2009 organizers for the transliteration datasets and the FIRE 2008 organizers for the CLIR datasets. We thank K Saravanan for his help in setting up and performing the CLIR experiments. Finally, we thank the anonymous reviewers for their thorough and diligent review comments, which have improved the quality of this paper significantly.

REFERENCES

Al-Onaizan, Y. and Knight, K. 2001. Translating named entities using monolingual and bilingual resources. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. 400–408.

Ballesteros, L. 2000. Cross language retrieval via transitive translation. In *In W.B. Croft (Ed.), Advances in information retrieval: Recent research from the CIIR*. Boston: Kluwer Academic Publishers., 203–234.

Brown, P. E., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics 19*, 263–311.

CLEF. 2007. Crosslingual evaluation forum.

Crammer, K. and Singer, Y. 2001. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research 3*, 2003.

FIRE. 2008. Forum for information retrieval evaluation.

Gollins, T. and Sanderson, M. 2001. Improving cross language retrieval with triangulated translation. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 90–95.

Goto, I., Kato, N., Uratani, N., and Ehara, T. 2003. Transliteration considering context information based on the maximum entropy method. In *Proceedings of MT-Summit IX*. 125132.

Jansche, M. and Sproat, R. 2009. Named entity transcription with pair n-gram models. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*. Association for Computational Linguistics, Suntec, Singapore, 32–35.

Jiampojamarn, S., Bhargava, A., Dou, Q., Dwyer, K., and Kondrak, G. 2009. Directl: a language independent approach to transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*. Association for Computational Linguistics, Suntec, Singapore, 28–31.

Jung, S. Y., Hong, S., and Paek, E. 2000. An english to korean transliteration model of extended markov window. In *Proceedings of the 18th conference on Computational linguistics*. 383–389.

Kang, B. J. and Choi, K. S. 2000. Automatic transliteration and back-transliteration by decision tree learning. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*. 1135–1411.

Knight, K. and Graehl, J. 1997. Machine transliteration. In *Computational Linguistics*. 128–135.

Lafferty, J. D., Mccallum, A., and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA, 282–289.

Lee, J. S. and Choi, K. S. 1998. English to korean statistical transliteration for information retrieval. In *Computer Processing of Oriental Languages*. 17–37.

Lehtokangas, R., Keskustalo, H., and Järvelin, K. 2008. Experiments with transitive dictionary translation and pseudo-relevance feedback using graded relevance assessments. *J. Am. Soc. Inf. Sci. Technol. 59,* 3, 476–488.

Li, H., Kumaran, A., , Zhang, M., and Pervouvhine, V. 2009. Whitepaper 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, Suntec, Singapore, 19–26.

Li, H., Kumaran, A., Pervouchine, V., and Zhang, M. 2009. Report of news 2009 machine transliteration shared task.

Malik, M. G. A., Boitet, C., and Bhattacharyya, P. 2008. Hindi urdu machine transliteration using finite-state transducers. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK, 537–544.

Mandl, T. and Womser-Hacker, C. 2004. How do named entities contribute to retrieval effectiveness? In *CLEF*. 833–842.

Mandl, T. and Womser-Hacker, C. 2005. The effect of named entities on effectiveness in cross-language information retrieval evaluation. In *SAC*. 1059–1064.

Nakov, P. and Ng, H. T. 2009. Improved statistical machine translation for resource-poor languages using related resource-rich languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 1358–1367.

Och, F. J. and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics 29,* 1, 19–51.

Oh, J.-h. and Choi, K.-s. 2002. An english-korean transliteration model using pronunciation and contextual rules. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*. 758–764.

Oh, J.-H., Uchimoto, K., and Torisawa, K. 2009. Machine transliteration using target-language grapheme and phoneme: Multi-engine transliteration approach. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*. Association for Computational Linguistics, Suntec, Singapore, 36–39.

Porter, M. F. 1980. An algorithm for suffix stripping. In *Program*. Vol. 14. 130–137.

Stalls, B. G. and Knight, K. 1998. Translating names and technical terms in arabic text. In *Proceedings of COLING/ACL Workshop on Computational Approaches to Semitic Languages*. 34–41.

Udupa, R., Jagarlamudi, J., and Saravanan, K. 2008. Microsoft research india at fire2008: Hindi-english cross-language information retrieval.

Udupa, R., Saravanan, K., Bakalov, A., and Bhole, A. 2009. They are out there, if you know where to look: Mining transliterations of oov query terms for cross language information retrieval. In *ECIR'09: Proceedings of the 31st European Conference on IR research on Advances in Information Retrieval*. Toulouse, France, 437–448.

Veeravalli, S., Yella, S., Pingali, P., and Varma, V. 2008. Statistical transliteration for cross language information retrieval using hmm alignment model and crf. In *Proceedings of the 2nd workshop on Cross Lingual Information Access (CLIA) Addressing the Information Need of Multilingual Societies*. 125132.

Wu, H. and Wang, H. 2007. Pivot language approach for phrase-based statistical machine translation. *Machine Translation 21,* 3, 165–181.

Xu, J. and Weischedel, R. 2005. Empirical studies on the impact of lexical resources on clir performance. *Inf. Process. Manage. 41,* 3, 475–487.

Zhou, Y., Huang, F., and Chen, H. 2008. Combining probability models and web mining models: a framework for proper name transliteration. *Inf. Technol. and Management 9,* 2, 91–103.