ISSN 2229-5518

Machine Transliteration for Indian Languages: A Literature Survey

Antony P J and Dr. Soman K P

Abstract— This paper address the various developments in Indian language machine transliteration system, which is considered as a very important task needed for many natural language processing (NLP) applications. Machine transliteration is an important NLP tool required mainly for translating named entities from one language to another. Even though a number of different transliteration mechanisms are available for worlds top level languages like English, European languages, Asian languages like Chinese, Japanese, Korean and Arabic, still it is an initial stage for Indian languages. Literature shows that, recently some recognizable attempts have done for few Indian languages like Hindi, Bengali, Telugu, Kannada and Tamil languages. This paper is intended to give a brief survey on transliteration for Indianlanguages.

Index Terms— Named Entity, Agglutinative, Natural Language Processing, Transliteration, Dravidian Languages

---

.

1 INTRODUCTION

Machine transliteration is the practice of transcribing a character or word written in one alphabetical system into another alphabetical system. Machin**e** transliteration can play an important role in natural language application such as information retrieval and machine translation, especially for handling proper nouns and technical terms, cross-language applications, data mining and information retrieval system. The transliteration model must be design in such a way that the phonetic structure of words should be preserve as closely as possible.

The topic of machine transliteration has been studied extensively for several different language pairs. Various methodologies have been developed for machine transliteration based on the nature of the languages considered. Most of the current transliteration systems use a generative model based on alignment for transliteration and consider the task of generating an appropriate transliteration for a given word. Such model requires considerable knowledge of the languages.

Transliteration usually depends on context. For example, the English (source) grapheme 'a' can be transliterated into Kannada (target) language graphemes on the basis of its context, like 'a', 'aa', 'ei' etc. Similarly 'i' can be transliterated either 'i' or 'ai' on the basis of its context. This is because vowels in English may correspond to long vowels or short vowels or some time combination of vowels [1] in Kannada during transliteration. Also on the basis of its context, consonants like 'c', 'd', 'l', or 'n', has multiple transliterations in Kannada language. For transliterating names, we have to exploit the phonetic correspondence of alphabets and sub-strings in English to Kannada. For example, "ph" and "f" both map to the same sound of (f). Likewise, "sha" in Kannada (as in Roshan) and "tio" in English (as in ration) sound similar. The transliteration model should be design while considering all these complexities.

1.1 Major Contribution to Machine Transliteration

The figure 1 shows the researchers who proposed different approaches to develop various machine transliteration systems.

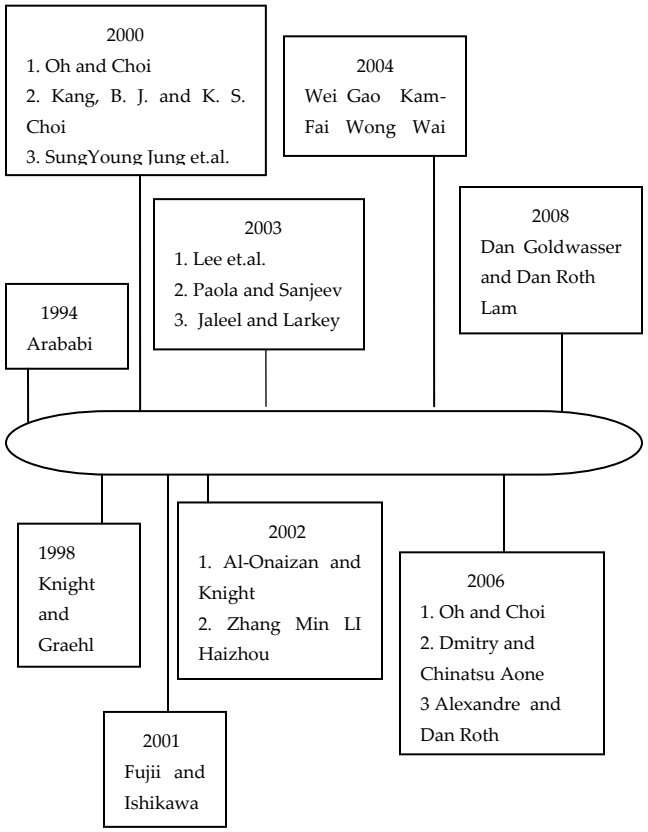

Fig.1. Contributors to Machine Transliteration

The very first attempt in transliteration was done by Arababi through a combination of neural network and expert systems for transliterating from Arabic-English in 1994 [2]. The proposed *neural network and knowledge-based hybrid system* generate multiple English spellings for Arabic person names.

The next development in transliteration was based on a *statistical based* approach proposed by Knight and Graehl in 1998 for back transliteration from English to Japanese Katakana. This approach was adapted by Stalls and Knight for back transliteration from Arabic to English.

There are three different machine transliteration developments in the year 2000, from three separate research team. Oh and Choi develped a phoneme based model using rule based approach incorporating phonetics as an intermediate representation. This English-Korean (E-K) transliteration model is built using pronunciation and contextual rules. Kang, B. J. and K. S. Choi, in their work presented an automatic character alignment method between English word and Korean transliteration. Aligned data is trained using supervised learning decision tree method to automatically induce transliteration and back-transliteration rules. This methodology is fully bidirectional, i.e. the same methodology is used for both transliteration and back transliteration. SungYoung Jung proposed a statistical English-to-Korean transliteration model that exploits various information sources. This model is a generalized model from a conventional statistical tagging model by extending Markov window with some mathemaical approximation techniques. An alignment and syllabification method is developed for accurate and fast operation.

In the year 2001, Fujii and Ishikawa describe a transliteration system for English-Japanese Cross Language Information Retrieval (CLIR) task that requires linguistic knowledge.

In the year 2002, Al-Onaizan and Knight developed a hybrid model based on phonetic and spelling mappings using Finite state machines. The model was designed for transliterating Arabic names into English. In the same year, Zhang Min LI Haizhou SU Jian proposed a direct orthographic mapping framework to model phonetic equivalent association by fully exploring the orthographical contextual information and the orthographical mapping. Under the DOM framework, a joint source-channel transliteration model (*n*-gram TM) captures the source-target word orthographical mapping relation and the contextual information.

An English-Arabic transliteration scheme was developed by Jaleel and Larkey based on HMM using GIZA++ approach in 2003. Mean while they also attempted to develop a transliteration system for Indian language. Lee et.al. [2003] developed the noisy channel model for English Chinese language pair, in which the back transliteration problem is solved by finding the most probable word E, given transliteration C. Letting $P(E)$ be the probability of a word E, then for a given transliteration C, the back-transliteration probability of a word E can be written as $P(E|C)$. This method requires no conversion of source words into phonetic symbols. The model is trained automatically on a bilingual proper name list via unsupervised learning. Model parameters are estimated using EM. Then, the channel decoder with Viterbi decoding algorithm is used to find the word $\hat{E}$ that is the most likely to the word E that gives rise to the transliteration C. The model is tested for English Chinese language pair. In the same year Paola Virga and Sanjeev Khudanpur demonstrated the application of statistical machine translation techniques to "translate" the phonemic representation of an English name, obtained by using an automatic text-to-speech system, to a sequence of initials and finals, commonly used subword units of pronunciation for Chinese.

Wei Gao Kam-Fai Wong Wai Lam proposed an efficient algorithm for phoneme alignment in 2004. In this a data driven technique is proposed for transliterating English names to their Chinese counterparts, i.e. forward transliteration. With the same set of statistics and algorithms, transformation knowledge is acquired automatically by machine learning from existing origin-transliteration name pairs, irrespective of specific dialectal features implied. The method starts off with direct estimation for transliteration model, which is then combined with target language model for postprocessing of generated transliterations. Expectation-maximization (EM) algorithm is applied to find the best alignment (Viterbi alignment) for each training pair and generate symbol-mapping probabilities. A weighted finite state transducer is built (WFST) based on symbol-mapping probabilities, for the transcription of an input English phoneme sequence into its possible pinyin symbol sequences.

Dmitry Zelenko and Chinatsu Aone proposed two discriminative methods for name transliteration in 2006. The methods correspond to local and global modeling approaches in modeling structured output spaces. Both methods do not require alignment of names in different languages but their features are computed directly from the names themselves. The methods are applied to name transliteration from three languages Arabic, Korean, and Russian into English. In the same year Alexandre Klementiev and Dan Roth developed a discriminative approach for translieration. A linear model is trained to decide whether a word T is a transliteration of an Named Entity S.

2 Machine Transliteration Approaches

Transliteration is generally classified in to three types namely, Grapheme based, Phoneme based, hybrid models and correspondence-based transliteration model [1][2].

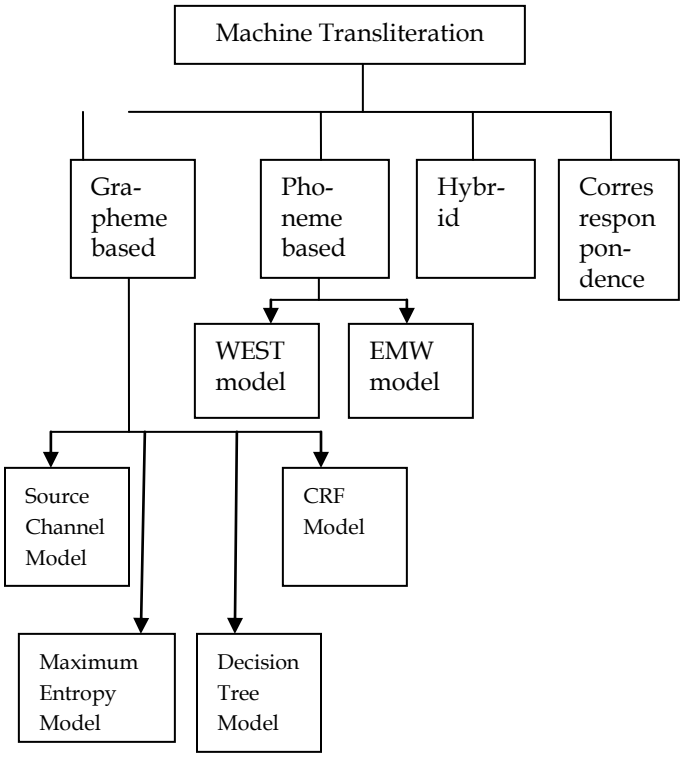

Fig. 2. General Classification of Machine Transliteration

These models are classified in terms of the units to be transliterated. The grapheme based approach (Lee & Choi, 1998; Jeong, Myaeng, Lee, & Choi, 1999; Kim, Lee, & Choi, 1999; Lee, 1999; Kang & Choi, 2000; Kang & Kim, 2000; Kang, 2001; Goto, Kato, Uratani, & Ehara, 2003; Li, Zhang, & Su, 2004) treat transliteration as an orthographic process and tries to map the source graphemes directly to the target graphemes. Grapheme based model is further divided in to (i) source channel model (iii) Maximum Entropy Model (iii) Conditional Random Field models and (iv) Decision Trees model. The grapheme-based transliteration model is sometimes referred to as the direct method because it directly transforms source language graphemes into target language graphemes without any phonetic knowledge of the source language words.

On the other hand, phoneme based models (Knight & Graehl, 1997; Lee, 1999; Jung, Hong, & Paek, 2000; Meng, Lo, Chen, & Tang, 2001) treat transliteration as a phonetic process rather than an orthographic process. Weighted Finite State Transducers (WFST) and extended Markov window (EMW) are the approaches belong to the phoneme based models. The phoneme-based transliteration model is sometimes referred to as the pivot method because it uses source language phonemes as a pivot when it produces target language graphemes from source language graphemes. This model therefore usually needs two steps: 1) produce source language phonemes from source language graphemes and 2) produce target language graphemes from source phonemes.

As the name indicates, a hybrid model (Lee, 1999; Al-Onaizan & Knight, 2002; Bilac & Tanaka, 2004) either use a combination of a grapheme based model and a phoneme based model or capture the correspondence between source graphemes and source phonemes to produce target language graphemes. Correspondence-based transliteration model was proposed by Oh & Choi, in the year 2002. The hybrid transliteration model and correspondence-based transliteration model make use of both source language graphemes and source language phonemes when producing target language transliterations. Figure 2 shows the general classification of machine transliteration system.

3 Machine Transliteration in India: A Literature Survey

According to Internet & Mobile Association of India (IAMAI), as on September 2008 India had 45.3 million active Internet users. The government of India has taken number of initiatives to enable rural Indians to access the Internet. This signifies need for providing information in regional languages to the user. Many technical terms and proper names, such as personal, location and organization names, are translated from one language into another language with approximate phonetic equivalents. The chapter is organized as follow: the first section gives a brief description of various approaches towards machine transliteration, followed by various transliteration attempts for Indian languages. The following sub sections describe the various machine transliteration developments in Indian languages.

3.1 English to Hindi Machine Transliteration

Literature shows that majority of work in machine transliteration for Indian languages were done in Hindi and Dravidian languages. The following are the noticeable developments in English to Hindi or other Indian languages to Hindi machine transliteration.

i) Transliteration as a Phrase Based Statistical Machine Translation: In 2009, Taraka Rama and Karthik Gali addressed the transliteration problem as translation problem [3]. They have used the popular phrase based SMT systems successfully for the task of transliteration. This is a stochastic based approach, where the publically available GIZA++ and beam search based decoder were used for developing the transliteration model. A well organized English- Hindi aligned corpus used to train and test the system. It was a prototype system and reported an accuracy of 46.3% on the test set.

ii) Another transliteration system was developed by Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay based on NEWS 2009 Machine Transliteration Shared Task training datasets [2]. The proposed transliteration system uses the modified joint source channel model along with two other alternatives to translate English to Hindi transliteration. The system also uses some post processing rules for the purpose of removing the errors in the system to improve the accuracy. They performed one standard run and two nonstandard runs in the developed English to Hindi transliteration system. The results showed that the performance of the standard run was better than the non standard one.

iii) Using the Letter- to- Phoneme technology, the transliteration problem was addressed by Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay in 2009 [2]. This approach was intended for improving the performance of the existing work with re-implementation using the specified technology. In the proposed system, transliteration problem is interpreted as a variant of the letter-to-phoneme (L2P) subtask of text to- speech processing. They apply a re-implementation of a state-of-the-art, discriminative L2P system to the problem, without further modification. In their experiment, they demonstrated that an automatic letter-to- phoneme transducer performs fairly well with no language specific or transliteration-specific modifications.

iv) An English to Hindi Transliteration using Context-Informed Phrase-based statistical machine translation (PB-SMT) was proposed by Rejwanul Haque, Sandipan Dandapat, Ankit Kumar Srivastava, Sudip Kumar Naskar and Andy Way CNGL in 2009 [2]. The transliteration system was modeled by translating characters rather than words as in character-level translation systems. They used

a memory-based classification framework that enables efficient estimation of these features while avoiding data sparseness problems. The experiments were both at character and transliteration unit (TU) level and reported that position - dependent source context features produce significant improvements in terms of all evaluation metrics. In this way the problem of machine transliteration was successfully implemented by adding source context modeling into state-of-the-art log-linear phrase-based statistical machine translation (PB-SMT). In their experiment, they also showed that by taking source context into account, improve the system performance substantially.

v) Abbas Malik, Laurent Besacier Christian Boitet and Pushpak Bhattacharyya proposed an Urdu to Hindi Transliteration using hybrid approach in 2009 [2]. This hybrid approach combines finite-state machine (FSM) based techniques with statistical word language model based approach and achieved better performance. The main effort of this system was to removal of diacritical marks from the input Urdu text. They report that the approach improved the system accuracy by 28.3% in comparison with their previous finite-state transliteration model.

vi) A Punjabi to Hindi transliteration system was developed by Gurpreet Singh Josan and Jagroop Kaur based on statistical approach in 2011 [1]. The system used letter to letter mapping as baseline and try to find out the improvements by statistical methods. They used a Punjabi – Hindi parallel corpus for training and publically available SMT tools for building the system.

3.2 English to Tamil Language Machine Transliteration

The first English to Tamil transliteration system was developed by Kumaran A and Tobias Kellner in the year 2007. Afraz and Sobha developed a statistical transliteration system using statistical approach in the year 2008. The third transliteration system was based on Compressed Word Format (CWF) algorithm and a modified version of Levenshtein's Edit Distance algorithm. Vijaya MS, Ajith VP, Shivapratap G, and Soman KP of Amrita University, Coimbatore proposed the remaining three English to Tamil Transliteration using different approaches.

i) Kumaran A and Tobias Kellner proposed machine transliteration framework based on a core algorithm modelled as a noisy channel, where the source string gets garbled into target string. Viterbi alignment is used for source and target language segments alignment. The transliteration is learned by estimating the parameters of the distribution that maximizes the likelihood of observing the garbling seen in the training data using Expectation Maximization algorithm. Subsequently, given a target language string 't', the most probable source language string s that gave raise to 't' is decoded. The method is applied for forward transliteration from English to Hindi, Tamil, Arabic, Japanese and backward transliteration from Hindi, Tamil, Arabic, Japanese to English.

ii) Afraz and Sobha developed a statistical transliteration engine using an n-grams based approach in the year 2008. This algorithm uses n-gram frequencies of the transliteration units, to find the probabilities. Each transliteration unit is pattern of consonant-vowel in the word. This transliteration engine is used in their Tamil to English cross language information retrieval (CLIR) system.

iii) Srinivasan C Janarthanam et.al. (2008) proposed an efficient algorithm for transliteration of English named entities to Tamil. In the first stage of transliteration process, he used a Compressed Word Format (CWF) algorithm to compress both English and Tamil named entities from their actual forms. Compressed Word Format of words is created using an ordered set of rewrite and remove rules. Rewrite rules replace characters and clusters of characters with other characters or clusters. Remove rules simply remove the characters or clusters. This CWF algorithm is used for both English and Tamil names, but with different rule set. The final CWF forms will only have the minimal consonant skeleton. In the second stage Levenshtein's Edit Distance algorithm is modified to incorporate Tamil characteristics like long-short vowel, ambiguities in consonants like 'n', 'r', 'i', etc. Finally the CWF Mapping algorithm is the transliteration algorithm that takes as input a source language named entity string, converts into CWF form, and maps with similar Tamil CWF words using modified edit distance and produces a ranked list of transliterated names in the target language Tamil.

iv) In the first attempt they have demonstrated a transliteration model for English to Tamil transliteration using Memory based learning by reformulating the transliteration problem as sequence labeling and multi classification in 2008 [4]. The proposed system was corpus based and they have used English- Tamil aligned parallel corpus of 30,000 person names and 30,000 place names to train the transliteration model. They evaluated the performance of the system based on top 5 accuracy and reported 84.16% exact English to Tamil transliteration.

v) In their second attempt, the transliteration problem was modeled as classification problem and trained using C4.5 decision tree classifier, in WEKA Environment [5]. The same parallel corpus was used to extract features and these features are used to train the WEKA algorithm. The resultant rules generated by the WEKA were used to develop the transliteration system. They reported exact Tamil transliterations for 84.82% of English names.

vi) The third English to Tamil Transliteration was developed using One Class Support Vector Machine algorithm in 2010 [6]. This is a statistical based transliteration system, where training, testing and evaluations were performed with publically available SVM tool. The experiment result

shows that, the SVM based transliteration was outperformed over other previous methods..

3.3 English to Kannada Language Machine Transliteration

Antony P J, Ajith VP, and Soman KP of Amrita University, Coimbatore proposed three different approaches for English to Kannada Transliteration. The proposed systems based on a well aligned bilingual parallel corpus of 40,000 English- Kannada place names.

i) The first proposed transliteration model is based on multi-class classification approach in which j48 decision tree classifier of WEKA was used for classification [7]. The parallel corpus consisting of 40,000 Indian place names was aligned properly and the extracted feature patterns were used to train the transliteration model. The accuracy of the model was tested with 1000 English names that were out of corpus. The model was evaluated by considering top 5 transliterations. The model was tested with a data set of 1000 English names and produced exact Kannada transliteration for English words with an accuracy of 81.25% when we considered only the top 1 result. We obtained an accuracy of 85.88% when we considered only the top 2 results. The overall accuracy is increased to 91.32%, when we considered top 5 results.

ii) The second method addresses the problem of transliterating English to Kannada language using Support Vector Machines (SVM) [8]. The proposed transliteration scheme uses sequence labeling method to model the transliteration problem. The framework was based on data driven method and one to one mapping approach. Which simplify the development procedure of transliteration system and facilities better improvement in transliteration accuracy while compared with other state-of-the-art machine learning algorithms. The model is trained on 40,000 words containing Indian place names. The model is evaluated by considering top 5 transliterations. From the experiment we found that considering first five ranked transliteration result increase the overall transliteration accuracy in a great extent. The system achieved exact Kannada transliterations for 87.28% of English names.

iii) The third English to Kannada transliteration system was developed using a publically available translation tool called Statistical Machine Translation (SMT) [9].The model is trained on 40,000 words containing Indian place names. During the training phase the model is trained for every class in order to distinguish between examples of this class and all the rest. The SVM binary classifier predicts all possible class labels for a given sequence of source language alphabets and selects only the most probable class labels. Also SVM generate a dictionary which consists of all possible class labels for each alphabet in the source language name. This dictionary avoids the excessive negative examples while training the model and training become faster. This transliteration technique was demonstrated for English to Kannada Transliteration and achieved exact Kannada transliterations for 89.27% of English names.

3.4 English to Malayalam Language Machine Transliteration

In the year 2009, Sumaja Sasidharan, Loganathan R, and Soman K P developed English to Malayalam Transliteration using Sequence labeling approach [10]. They have used a parallel corps consisting of 20000 aligned English-Malayalam person names for training the system. The approach is very similar to earlier English to Tamil transliteration. The model produced the Malayalam transliteration of English words with an accuracy of 90% when tested with 1000 names.

3.5 English to Telugu Language Machine Transliteration

An application of transliteration was proposed by V.B. Sowmya and Vasudeva Varmain 2009 [11]. They proposed a transliteration based text input method for Telugu, in which the user's type Telugu using Roman script using simple edit-distance based approach. They have tested the approach with three datasets – general data, countries and places and person names and reported the performance of the system.

3.6 English to Indian Language Machine Transliteration

A well known on line transliteration system for Indian language is Google Indic transliteration which works reasonable well for English to Indian languages. There are also Keyboard layouts like Inscript and Keylekh transliteration that have been available for Indian languages. The following are the generic approach for machine transliteration for English to Indian languages.

i) Harshit Surana and Anil Kumar Singh in 2008, proposed a transliteration system using two different methods on two Indian languages Hindi and Telugu [12]. In their experiment, using character based n-grams, a word is classified into two classes either Indian or foreign. The proposed technique considered the properties of the scripts but does not require any training data on the target side, while it uses more sophisticated techniques on the source side. The proposed model first identifies the class of the source side word to identify whether foreign or Indian word. Based on the identified class, the system uses any one of the two methods. The system uses the easily creatable mapping tables and a fuzzy string matching algorithm to get the target word.

ii) Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay proposed a transliteration technique based on orthographic rules and phoneme based approach and system was trained on the NEWS 2010 transliteration datasets [13]. In their experiments, they have submitted one standard run and two non-standard runs were submitted for English to Hindi and Bengali translite-

ration while one standard and one non-standard run were submitted for Kannada and Tamil. The reported results were as follow: For the standard run, the system demonstrated means F-Score values of 0.818 for Bengali, 0.714 for Hindi, 0.663 for Kannada and 0.563 for Tamil. The reported mean F-Score values of non-standard runs are 0.845 and 0.875 for Bengali non-standard run-1 and 2, 0.752 and 0.739 for Hindi non-standard run-1 and 2, 0.662 for Kannada non-standard run-1 and 0.760 for Tamil non-standard run-1. Non-Standard Run-2 for Bengali has achieved the highest score among all the submitted runs. Hindi Non-Standard Run-1 and Run-2 runs are ranked as the 5th and 6th among all submitted Runs.

iii) K Saravaran, Raghavendra Udupa and A Kumaran proposed a cross-lingual information retrieval system enhanced with transliteration generation and mining in 2010 [14]. They proposed Hindi-English and Tamil-English cross-lingual evaluation tasks, in addition to the English-English monolingual task. They used a language modeling based approach using query likelihood based document ranking and a probabilistic translation lexicon learned from English-Hindi and English-Tamil parallel corpora. To deal with out-of-vocabulary terms in the cross-lingual runs, they proposed two specific techniques. The first technique is to generate transliterations directly or transitively, and second technique is to mining possible transliteration equivalents from the documents retrieved in the first-pass. In their experiment they showed that both of these techniques significantly improved the overall retrieval performance of our cross-lingual IR system. The systems achieved a peak performance of a MAP of 0.4977 in Hindi-English and 0.4145 in the Tamil-English.

iv) DLI developed a unified representation for Indian language called an Om transliteration which is similar to to ITRANS (Indian language Transliteration scheme) [15]. To enhance the usability and readability, Om has been designed on the following principles: (i) easy readability (ii) case-insensitive mapping and (iii) phonetic mapping, as much as possible. In Om transliteration system, when a user is not interested in installing language components, or when the user cannot read native language script the text may be read in English transliteration itself. Even in the absence of Om to native font converters, people around the globe can type and publish texts in the Om scheme which can be read and understood by many, even when they cannot read the native script.

v) Using statistical alignment models and Conditional Random Fields (CRF), a language independent transliteration system was developed by Shishtla, Surya Ganesh V, Sethuramalingam Subramaniam and Vasudeva Varma in 2009 [2]. Using the expectation maximization algorithm, statistical alignment models maximizes the probability of the observed (source, target) word pairs and then the character level alignments are set to maximum posterior predictions of the model. The advantage of the system is that no language-specific heuristics used in any of the modules and hence it is extensible to any language-pair with least effort.

vi) Using a phrase-based statistical machine translation approach to an English-Hindi, English-Tamil and English-Kannada transliteration system was developed by Manoj Kumar Chinnakotla and Om P. Damani in 2009 [2]. In the proposed SMT based system, words are replaced by characters and sentences by words and GIZA++ was used for learning alignments and Moses for learning the phrase tables and decoding. In addition to standard SMT parameters tuning, the system also focus on tuning the Character Sequence Model (CSM) related parameters like order of the CSM, weight assigned to CSM during decoding and corpus used for CSM estimation. The results show that improving the accuracy of CSM pays off in terms of improved transliteration accuracies.

vii) Kommaluri Vijayanand, Inampudi Ramesh Babu and Poonguzhali Sandiran proposed the transliteration systems for English to Tamil language based on the reference corpora consists of language pair of 1000 names in 2009 [2]. The proposed transliteration system was implemented using JDK 1.6.0 for transliterating the English Named Entities in to Tamil language. From the experiment they found that the accuracy in top-1 score of the system was 0.061.

viii) Transliteration between Indian languages and English using an EM algorithm was proposed by Dipankar Bose and Sudeshna Sarkar in 2009 [2]. They used an EM algorithm to learn the alignment between the languages. They found that there is lot of ambiguities in the rules mapping the characters in the source language to the corresponding characters in the target language. They handled some of these ambiguities by capturing context by learning multi-character based alignments and use of character n-gram models. They have used multiple models and a classifier to decide which model to use in their system. Both the models and classifiers are learned in a completely unsupervised manner. The performance of the system was tested for English and several Indian languages. They have used an additional preprocessor for Indian languages, which enhances the performance of the transliteration model. One more advantage is that, the proposed system is robust in the sense that it can filter out noise in the training corpus, can handle words of different origins by classifying them into different classes.

ix) Using word-origin detection and lexicon lookup method, an improvement in transliteration was proposed by Mitesh M. Khapra and Pushpak Bhattacharyya in 2009 [2]. The proposed improved model uses the following framework: (i) a word-origin detection engine (*pre-processing*)

(ii) a CRF based transliteration engine and (iii) a re-ranking model based on lexiconlookup (*post-processing*). They applied their idea on *English-Hindi* and *English-Kannada* transliteration and reported 7.1% improvement in top-1 accuracy. The performance of the system was tested against the NEWS 2009 dataset. They submitted one standard run and one non-standard run for the English-Hindi task and one standard run for the English-Kannada task.

x) Sravana Reddy and Sonjia Waxmonsky proposed a substring-based transliteration with conditional random Fields for English to Hindi, Kannada and Tamil languages in 2009 [2]. The proposed transliteration system was based on the idea of phrase-based machine translation. In the transliteration system, phrases correspond to multi-character substrings. So, source and target language strings are treated not as sequences of characters but as sequences of non-overlapping substrings in the proposed system. Using Conditional Random Fields (CRFs), they modeled the transliteration as a 'sequential labeling task' where substring tokens in the source language are labeled with tokens in the target language. The system uses both 'local contexts' and 'phonemic information' acquired from an English pronunciation dictionary. They evaluated the performance of the system separately for Hindi, Kannada and Tamil languages using a CRF trained on the training and development data, with the feature set U+B+T+P.

xi) Balakrishnan Vardarajan and Delip Rao proposed an ε-extension Hidden Markov Models and Weighted Transducers for Machine Transliteration from English to five different languages, including Tamil, Hindi, Russian, Chinese, and Kannada in 2009 [2]. The developed method involves deriving substring alignments from the training data and learning a weighted finite state transducer from these alignments. They have defined a ϱ-extension Hidden Markov Model to derive alignments between training pairs and a heuristic to extract the substring alignments. The performance of the transliteration system was evaluated based on the standard track data provided by the NEWS 2009. The main advantage of the proposed approach is that the system is language agnostic and can be trained for any language pair within a few minutes on a single core desktop computer.

xii) Raghavendra Udupa, K Saravanan, A Kumaran and Jagadeesh Jagarlamudi addressed the problem of mining transliterations of Named Entities (NEs) from large comparable corpora in 2009 [2]. They have proposed a mining algorithm called Mining Named-entity Transliteration equivalents (MINT), which uses a cross-language document similarity model to align multilingual news articles and then mines NETEs from the aligned articles using a transliteration similarity model. The main advantage of MINT is that, it addresses several challenges in mining NETEs from large comparable corpora: exhaustiveness (in mining sparse NETEs), computational efficiency (in scaling on corpora size), language independence (in being applicable to many language pairs) and linguistic frugality (in requiring minimal external linguistic resources). In their experiment they showed that the performance of the proposed method was significantly better than a state-of-the-art baseline and scaled to large comparable corpora.

xiii) Rohit Gupta, Pulkit Goyal and Sapan Diwakar proposed a transliteration system among Indian languages using WX Notation in 2010 [16]. They have proposed a new transliteration algorithm which is based on Unicode transformation format of an Indian language. They tested the performance of the proposed system on a large corpus having approximately 240k words in Hindi to other Indian languages. The accuracy of the system is based on the phonetic pronunciations of the words in target and source language and this was obtained from Linguistics having knowledge of both the languages. From the experiment, they found that the time efficiency of the system is better and it takes less than 0.100 seconds for transliterating 100 Devanagari (Hindi) words into Malayalam when run on an Intel Core 2 Duo, 1.8 GHz machine in Fedora.

xiv) A grapheme-based model was proposed by Janarthanam, Sethuramalingam and Nallasamy in 2008 [2]. In this proposed system, the transliteration equivalents are identified by matching in a target language database based on edit distance algorithm. The transliteration system was trained with several names and then the trained model is used to transliterate new names.

xv) In a separate attempt, Surana and Singh proposed another algorithm for transliteration in 2008 that eliminates the training phase by using fuzzy string matching approach [2].

4 Conclusion

In this paper work, we have presented a survey on developments of different machine transliteration systems for Indian languages. Additionally we tried to give a brief idea about the existing approaches that have been used to develop machine transliteration tools. From the survey I found out that almost all existing Indian language machine transliteration systems are based on statistical and hybrid approach. The main effort and challenge behind each and every development is to design the system by considering the agglutinative and morphological rich features of language.

5 Acknowledgments

We acknowledge our sincere gratitude to Mr. Benjamin Peter (Assistant Professor, MBA Dept, St.Joseph Engineering College, Mangalore, India) and Mr. Rakesh Naik (Assistant Professor, MBA Dept, St.Joseph Engineering College, Mangalore, India) for their valuable support regarding proof reading and correction of this survey paper.

6 REFERENCES

[1] Gurpreet Singh Josan & Jagroop Kaur (2011)' Punjabi to Hindi Statistical Machine Translaiteration', International Journal of Information Technology and Knowledge Management July-December 2011, Volume 4, No. 2, pp. 459-463.

[2] Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay (2009), 'English to Hindi Machine Transliteration System at NEWS', Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, page 80-83, Suntec, Singapore.

[3] Taraka Rama, Karthik Gali (2009), 'Modeling Machine Transliteration as a Phrase Based Statistical Machine Translation Problem', Language Technologies Research Centre, IIIT, Hyderabad, India.

[4] Vijaya MS, Ajith VP, Shivapratap G, and Soman KP (2008), 'Sequence labeling approach for English to Tamil Transliteration using Memory based learning', In Proceedings of Sixth International Conference on Natural Language processing.

[5] Vijaya MS, Ajith VP, Shivapratap G, and Soman KP (2009), 'English to Tamil Transliteration using WEK',International Journal of Recent Trends in Engineering, Vol. 1, No. 1.

[6] Vijaya M. S., Shivapratap G., Soman K. P (2010), 'English to Tamil Transliteration using One Class Support Vector Machine', International Journal of Applied Engineering Research, Volume 5, Number 4, 641-652.

[7] Antony P J, Ajith V P and Soman K P (2010), 'Feature Extraction Based English to Kannada Transliteration', Third International Conference on Semantic E-business and Enterprise Computing, SEEC-2010.

[8] Antony P J, Ajith V P and Soman K P (2010), 'Kernel Method for English to Kannada Transliteration', International Conference on-Recent Trends in Information, Telecommunication and Computing (ITC 2010), Paper is archived in the IEEE Xplore and IEEE CS Digital Library.

[9] Antony P J, Ajith V P and Soman K P (2010), 'Statistical Method for English to Kannada Transliteration', Lecturer Notes in Computer Science-Communications in Computer and Information Science (LNCS-CCIS), Volume 70, 356-362, DOI: 10.1007/978-3-642-12214-9_57.

[10] Sumaja Sasidharan, Loganathan R, and Soman K P (2009), 'English to Malayalam Transliteration using Sequence labeling approach',. International Journal of Recent Trends in Engineering, Vol. 1, No. 2.

[11] V.B. Sowmya, Vasudeva Varma in (2009), 'Transliteration Based Text Input Methods for Telugu', 22and International Conference on Computer Processing for Oriental Languages" (ICCPOL-2009), Hongkong.

[12] Harshit Surana and Anil Kumar Singh, 'A More Discerning and Adaptable Multilingual Transliteration Mechanism for Indian Languages', Language Tech. Research Centre IIIT, Hyderabad, India.

[13] Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay (2010), 'A transliteration technique based on orthographic rules and phoneme based approach', NEWS 2010.

[14] K Saravaran Raghavendra Udupa, A Kumaran (2010), 'Crosslingual Information Retrieval System Enhanced with Transliteration Generation and Mining', Microsoft Research India, Bangalore , India.

[15] Prashanth Balajapally, Phanindra Bandaru, Madhavi Ganapathiraju N. Balakrishnan and Raj Reddy , 'Multilingual Book Reader: Transliteration, Word-to-Word Translation and Full-text Translation'.

[16] Rohit Gupta, Pulkit Goyal and Sapan Diwakar (2010), 'Transliteration among Indian Languages using WX Notation by. Semantic Approaches in Natural Language Processing', Proceedings of the Conference on Natural Language Processing 2010. Pages 147-151, Universaar, Universitätsverlag des Saarlandes Saarland University Press, Presses universitaires de la Sarre.