SURVEY ON MACHINE TRANSLITERATION AND MACHINE LEARNING MODELS

M L Dhore[1],R M Dhore[2] and P H Rathod[3]

[1,3]Vishwakarma Institute of Technology, Savitribai Phule Pune University, India
[2]Pune Vidhyarthi Girha's College of Engineering and Technology, SPPU, India

ABSTRACT

Globalization and growth of Internet users truly demands for almost all internet based applications to support local languages. Support of local languages can be given in all internet based applications by means of Machine Transliteration and Machine Translation. This paper provides the thorough survey on machine transliteration models and machine learning approaches used for machine transliteration over the period of more than two decades for internationally used languages as well as Indian languages. Survey shows that linguistic approach provides better results for the closely related languages and probability based statistical approaches are good when one of the languages is phonetic and other is non-phonetic.Better accuracy can be achieved only by using Hybrid and Combined models.

KEYWORDS

CRF, Grapheme, HMM, Machine Transliteration, Machine Learning, NCM, Phoneme, SVM

1.INTRODUCTION

Machine Transliteration (MT) has received significant research attention in recent years. Given a source term, machine transliteration refers to generating its phonetic equivalent in the target language for Out Of Vocabulary (OOV) words.OOV words mainly consist of named entities which include person/location/organization names and technical terms. The reverse process is known as Backward Transliteration. Named Entity transliteration is required in many applications which include cross-language information retrieval, corpus alignment, information extraction, machine translation, and automatic lexicon acquisition[1-2].This paper presents a review of previous work carried out by the researchers related with transliteration generation over the last two decades.Table 1 depicts the language pairs used at international level for MT.

Table 1. Language Pairs Used For Machine Transliteration

English - Russian	English - Chinese	English - Hindi	English - Japanese Katakana
English - Pinyin	Japanese - English	Urdu - English	Shahmukhi - Gurmukhi
Chinese - English	Pinyin - Chinese	English - Oriya	English - Korean Hangul
English - Thai	English - Arabic	English-Punjabi	English - Kannada
Spanish - Chinese	English - Japanese	English -Telugu	Bengali - English
English - Hebrew	Arabic to French	Hindi - English	Spanish - English
English - Korean	English – Spanish	Punjabi-English	Swedish - Finnish
Thai - English	Persian - English	English - Tamil	Arabic - English

2. EXISTING MODELS FOR NAMED ENTITY MACHINE TRANSLITERATIONS

There are mainly four models are being used for the machine transliteration.

2.1. Grapheme/Spelling based Transliteration Model

This model considers transliteration as an orthographic process and maps the source language graphemes/character/characters directly to the target language graphemes/character/characters. Theoretically, it is a direct orthographical mapping from source graphemes/character/characters to target graphemes/character/characters. This model is also sometimes referred to as the direct/spelling method as it directly transforms source language graphemes/character/characters into target language graphemes/character/characters without any phonetic knowledge of the source language words.

2.2. Phoneme based Transliteration Model

This model considers transliteration as a phonetic process rather than an orthographic process. In this model, transliteration process is treated as a conversion from source graphemes/character/characters to source phoneme/phonetic followed by a conversion from source phoneme/phonetic to target graphemes/character/characters. For this model, the transliteration key is pronunciation or the source phoneme/phonetic rather than spelling or the source phoneme/phonetic. This model is basically a source graphemes/character/characters to source phoneme/phonetic transformation and source phoneme/phonetic to target graphemes/character/characters transformation.

2.3. Hybrid based Transliteration Model

This model simply combines Grapheme based model and Phoneme based model through linear interpolation. It combines the grapheme based transliteration probability and the phoneme based transliteration probability using linear interpolation.

2.4. Combined/Correspondence based Transliteration Model

This model combines any number of the grapheme or phoneme based methods but not both [3].

3. EXISTING MACHINE LEARNING APPROACHES

There are mainly two approaches being used for the machine learning, Rule or Linguistic based and Statistical based Machine Learning

3.1 Rule/Linguistic based Machine Learning

The linguistic approach generally uses rules manually written by linguists and other heuristics to classify the named entities. The linguistic approach uses hand crafted rules based on pattern matching which need a linguistic analysis to formulate rules. It requires an advanced knowledge of grammar and other language related rules. This approach demands thorough knowledge and advanced skills related to the language under consideration. Table 2 shows the machine transliteration carried out using Linguistic Approach.

Table 2. Machine Transliteration Using Linguistic Approach

Author, Year	Language Pair	MT Model, Approach, Learning Model
Arbabi et al.(1994)	Arabic-English	Phoneme,Linguistic, Handcrafted Rules
Wan et al.(1998)	English-Chinese	Phoneme, Linguistic, syllabification
Jung et al.(2000)	English-Korean	Phoneme, Linguistic, Extended Markov
Oh et al.(2002)	English-Korean	Phoneme, Linguistic,, Contextual Rules
Jaleelet al.(2003)	English-Arabic	Grapheme, Hybrid, Rules and Bi-grams
Malik et al.(2006)	Shahmukhi-Gurmukhi	Grapheme, Linguistic, Handcrafted Rules
Mandal(2007)	Bengali - English	Phoneme, Linguistic, Character Mapping
Suranaet al.(2008)	English-Hindi/Telugu	Phoneme, Linguistic, DATM
Sahaet al.(2008)	Hindi/Bengali-English	Phoneme, Linguistic, Handcrafted Rules
Vijayanand(2009)	English-Tamil	Phoneme, Linguistic, Handcrafted Rules
Vijaya et al.(2009)	English-Tamil	Grapheme, Linguistic, Handcrafted Rules
Chai et al.(2010)	English-Thai	Grapheme, Linguistic, Syllabification
Josan et al.(2010)	Punjabi-Hindi	Grapheme, Linguistic, Character Mapping
Deep et al.(2011)	Punjabi - English	Grapheme, Linguistic, Character Mapping
Ben et al.(2011)	Arabic - French	Grapheme, Linguistic, Rule Oriented
Dhore et al.(2012)	Marathi/Hindi-English	Phoneme, Linguistic, Stress Analysis
Dhore et al.(2012)	Hindi/Marathi-English	Phoneme, Linguistic, Statistical, DDDM
Bhalla et al.(2013)	English-Panjabi	Grapheme, Linguistic, Syllabification

3.2 Statistical based Machine Learning Approach

The statistical based models uses a statistical learning approach which tries to generate the transliterations based on the probability statistics obtained from the bilingual corpora.

4. MACHINE LEARNING MODELS

4.1 Noisy Channel Model (NCM)

The basic Phrase Based Statistical Machine Translation (PB-SMT) model is an instance of the noisy channel approach in which the translation of a French sentence f into an English sentence e is modeled. This model was developed for machine translation where the input was a French sentence by *Peter E Brown et al. in 1993*. The same analogy further continued for *Statistical Machine Transliteration* (SMT) by replacing a word in the sentence by a character or a group of characters in the named entities. Brown's mathematical modeling of translation is further extended for transliteration [4]. Table 3 shows the machine transliteration carried out using NCM and SMT.

Table 3. Machine Transliteration Using NCM

Author, Year	Language Pair	MT Model, Approach, Learning Model
Yan et al.(2003)	English-Japanese	Phoneme,Statistical, Binary Validation
Lee et al.(2003)	English-Chinese	Grapheme, Statistical, SMT and EM
Hermjakob et al(2008)	Arabic - English	Grapheme, Statistical, SMT
Finch et al.(2009)	English-Japanese	Grapheme, Statistical, PB-SMT
Yuxiang et al.(2009)	English-Chinese	Grapheme, Statistical, NCM
Paul et al.(2009)	Spanish - English	Grapheme, Statistical, PB-SMT
Rama et al.(2009)	English-Hindi	Grapheme, Statistical, NCM
Kaur et al.(2011)	English-Punjabi	Grapheme, Statistical, SMT and Rule Based
Josanet al.(2011)	Punjabi-Hindi	Grapheme, Statistical, NCM
Sharma et al.(2012)	English-Hindi	Grapheme, Statistical, PB-SMT
Kumar et al.(2013)	Punjabi - English	Grapheme, SMT, N-gram
Joshi et al.(2013)	English-Hindi	Grapheme, SMT,Syllabification

4.2 Source Channel Model (SCM)

This is a mixed model which borrows concepts from both the rule based and statistical approaches. Based on Bayes Theorem, it describes a generative model. This model is implemented by *Kevin Knight et al. in 1998*[5].Table 4 shows the machine transliteration carried out using Source Channel Model (SCM).

Table 4. Machine Transliteration Using SCM

Author, Year	Language Pair	MT Model, Approach, Learning Model
Knight et al.(1998)	English-Japanese	Phoneme, Statistical, WFST and SCM
Lee et al.(1998)	English-Korean	Grapheme, Statistical, SCM
Stalls et al.(1998)	Arabic - English	Phoneme, Statistical, WFST
Al-Onaizan(2002)	Arabic-English	Hybrid, SCM and WFST
Gao et al.(2004)	English-Chinese	Phoneme, Statistical, SCM
Bilac et al.(2005)	Japanese-English	Hybrid, SCM and EM,WFST
Karimi et al.(2008)	English↔Persian	Combined, SCM and Voted Method

4.3 Joint Source Channel Model (JSCM)

This model is proposed by *Li Haizhou, Zhang Min and Su Jian in 2004*. Li Haizhou et al. have given the following mathematical modeling of JSCM. The source channel model represents the conditional probability of target names given a source name P(T|S). Unlike the noisy channel model, the joint source channel model does not try to capture how source names can be mapped to target names, but rather how source and target names can be generated simultaneously. In other words, it estimates a joint probability model that can be easily marginalized in order to yield conditional probability models for both forward transliteration and back-transliteration [6]. Table 5 shows the MT carried out using JSCM.

Table 5. Machine Transliteration Using JSCM

Author, Year	Language Pair	MT Model, Approach, Learning Model
Li et al.(2004)	English-Chinese.	Grapheme, Statistical, JSCM
Yang et al.(2009)	Japanese-Japanese Kanji	Grapheme, Statistical, JSCM and CRF
Das et al.(2009)	English-Hindi	Grapheme, Statistical, JSCM
Chen et al.(2011)	English↔ Chinese	Grapheme, Statistical, JSCM

4.4 Hidden Markov Model (HMM)

The HMM is a probabilistic function of Markov process. Markov model was first developed by *Andrei A Markov in 1913* for modeling the letter sequences in Russian literature. Mathematical model of HMM is described in the paper of *L Rabiner et al.1989* for the speech recognition and then after it is extended for translation and transliteration. The Hidden Markov Model (HMM) is one of the powerful statistical probability tool for modeling generative sequences that can be characterized by an underlying process generating an observable sequence [7-8].Table 6 shows the MT carried out using Hidden Markov Model (HMM).

Table 6 . Machine Transliteration Using HMM

Author, Year	Language Pair	MT Model, Approach, Learning Model
Jeong et al.(1999)	Korean-English	Phoneme, Statistical, HMM
Kang et al.(2000)	English-Korean	Grapheme, Statistical, HMM
Ganesh et al.(2008)	English-Hindi	Grapheme, Statistical, HMM and CRF
Kondrak et al.(2009)	English To Chinese	Grapheme, Statistical, HMM, N-Gram
Peter et al. (2009)	English-Russian	Grapheme, Statistical, HMM and WFST
Vardarajan(2009)	English-Tamil/Hindi	Grapheme, Statistical, HMM and WT
Zhou et al.(2009)	English↔ Pinyin	Grapheme, Statistical, HMM, N-Gram,

4.5 Conditional Random Fields (CRF)

This model is given by *J. Lafferty, A. McCallum, and F. Pereira in 2001*. They presented conditional random fields, a framework for building probabilistic models to segment and label sequence data. The mathematical model for CRF is described in the paper of John Lafferty et al. (2001) to segment and label sequence data. A CRF is a form of an undirected graphical model. CRF defines a single log-linear distribution over label sequences given a particular observation sequence. CRF model defines a conditional probability denoted as P (Y|X) over label sequences given a particular observation sequence X and does not calculate a joint distribution over both label and observation sequence [9-11]. Table 7 shows the MTby CRF.

Table 7.Machine Transliteration Using CRF

Author, Year	Language Pair	MT Model, Learning Model
Ganesh et al.(2008)	English-Hindi	Grapheme, Statistical, CRF and HMM
Oh et al.(2009)	English-Chinese/Hindi	Grapheme, Statistical, CRF, MIRA,EM
Reddy et al.(2009)	English To Hindi, Tamil	Phoneme, Statistical, CRF
Yang et al.(2009)	Japanese-Japanese Kanji	Grapheme, Statistical, CRF and JSCM
Ying et al.(2011)	English ↔ Chinese	Hybrid, Statistical, Combined CRFs
Jiang et al. (2011)	English-Chinese	Grapheme, Statistical, CRF
Waleed et al. (2012)	Arabic-English	Grapheme, Statistical, CRF
Dhoreet al.(2012)	Hindi/Marathi-English	Grapheme, Statistical, CRF

4.6 Maximum Entropy Model (MEM)

This model is proposed by *A L Berger, S D Pietra, and V J Della Pietra in 1996* for NLP. The maximum entropy model (MEM) is a probability based model which incorporates heterogeneous information effectively. Mathematical model for MEM is described in the paper of *A L Berger* et al. (1996) [12]. Table 8 shows the MT carried out using MEM.

Table 8. Machine Transliteration Using MEM

Author, Year	Language Pair	MT Model,Learning Model
Oh et al.(2006)	English To Korean, Japanese	Hybrid, MEM and MBL
Oh et al.(2007)	English To Korean, Japanese	Combined, MEM and SVM

4.7 Support Vector Machine (SVM)

This model is first introduced by *Bernhard E Boser, Isabelle M Guyon and Vladimir N Vapnik in 1992*. This model is represented as statistical learning theory by *Vladimir N Vapnik in 1999*. SVM does the classification by constructing an n-dimensional hyperplane which optimally segregates the data into two partitions. SVM is a new avatar of kernel functions with a supervised learning approach. It learns from a set of inputs values with the associated output values. It constructs a hyperplane between two classes using binary classifier. It is a binary classifier in which data points are classified in two classes with +1 and -1 labels. While separating input examples in two classes it maximise the separation between two classes using the method called as max margin. Due to max margin separation error rate gets minimised and if any new input with unknown label arrives for classification, the chances of making error is minimised [13].Table 9 shows the MT carried out using SVM.

Table 9. Machine Transliteration Using SVM

Author, Year	Language Pair	MT Model, Learning Model
Sumaja et al.(2009)	English To Malayalam,	Phoneme, Statistical, SVM
Oh et al.(2007)	Korean, Japanese	Combined, SVM and MEM
Antony et al.,(2010)	English-Kannada	Phoneme, Statistical, SVM
Kishorjit et al.(2012)	Bengali-Meitei Mayek	Grapheme, Statistical, SVM
Rathod et al.(2013)	Hindi, Marathi To English	Grapheme, Statistical, SVM

4.8 Decision Trees (DT)

This model is proposed by Ross Quinlan in 1970. In decision tree approach a tree like model of decisions is used along with their possible outcomes. These possible outcomes could be chance event outcomes, resource costs, and utility. It is a method to depict an algorithm in which decision tree is used in decision analysis that is to identify a mechanism which is most likely to reach a goal. Decision tree model is used along with probability models where decisions are to be taken runtime with no recall under incomplete knowledge. Decision tree is used to describe calculations of conditional probabilities [14].Table 10 shows the MT carried out using DT.

Table 10. Machine Transliteration Using DT

Author, Year	Language Pair	MT Model, Learning Model
Kang et al.(2000)	English-Korean	Grapheme, Statistical, Decision Trees
Oh et al.(2006)	English To Korean&Japanese	Hybrid, MEM, MBL andDT

5. REVIEW OF RULE BASED AND GRAPHEME BASED MODELLING

Lee J S and Choi K S (1998) developed their systems with direct orthographical mapping from source graphemes to target graphemes. They used the *source channel model*for English to Korean transliteration. They used a chunk of graphemes which corresponds to a source phoneme. First of all, English words were segmented into a chunk of English graphemes. Secondly, they produced possible chunks of Korean graphemes corresponding to the chunk of English graphemes. Finally,

the most relevant sequence of Korean graphemes was identified by using the source channel model. The key advantage of this technique is that, it considered a chunk of graphemes to represent a phonetic property of the source language word. However, errors propagating from first step of segmentation of the English word make it difficult to produce correct transliterations in further forwarding steps. Their approach has high time complexity due to the all possible chunks generation[15].*Corpus=1700, Training=1500, Testing=300, Word Accuracy (WA) = 63.3%, Character Accuracy (CA) = 78.5%*

Kang I H and Kim G (2000)proposed a method for English-Korean forward transliteration and back-transliteration. First, they performed English to Korean by using direct and pivot method and then they performed transliteration and back-transliteration using phoneme chunks. In the pivot method, transliteration was done in two steps, converting English words into pronunciation symbols and then converting these symbols into Korean words by using the Korean standard conversion rule. In the direct method, English words were directly converted to Korean words without intermediate steps. They used the *statistical transliteration approach* for transliteration mapping for their language model and they used the following bigramapproach[16]. *Dataset and Results: Corpus Set-I = 1650 and Set-II = 7185, For Set-I WA: 55.3% for Top-1 and 34.7% for* ***b****ack-****t****ransliteration (****bt****) and For Set-II, WA= 58.3% and 40.9% for bt.*

Kang B J and Choi K S (2000) developed English to Korean forward transliteration and backward transliteration system using *decision tree* learning. In their method decision trees were used for learning and to transform each source grapheme into target graphemes.This approach was considered the left three and the right three contexts and not any phonetic aspects of transliteration. The 26 decision trees were learned for each English letter and 46 decision trees were learned for each Korean letters [17]. *Results: Corpus=7000, Training=6000, Testing=1000, WA= 44.9% for left to right context and 34.2% for back-transliteration.*

Kang B J and Choi K S (2001) implemented the two approaches , transliteration and back-transliteration approach, and compared their relative effectiveness in Korean information retrieval. In the transliteration approach foreign words and English words were extracted and then English words were transliterated into Korean phonetic equivalents . Finally, they measured phonetic similarities between foreign words and equivalence classes were constructed .In the back-transliteration approach, first foreign words and English words were extracted and then foreign words were back-transliterated into their origin English word. Lastly, they measured phonetic similarities between English strings, equivalence classes are constructed[18]. *Corpus=7000, Training=6000, Testing=1000, WA= 51.3% and 37.2% for bt.*

Isao Goto, Naoto Kato, Noriyoshi Uratani and Terumasa Ehara(2003)proposed a method based on a transliteration network for English to Japanese transliteration. Transliteration method generated a Japanese katakana word from OOV English words which were not available in bilingual corpus and pronunciation dictionaries. For all such OOV words, an English word was divided into transliteration conversion units. These conversion units were partial English character strings in an English word. Then this conversion unit was converted into a partial katakana character string. To produce an adequate transliteration, they applied three approaches .First approach calculated the likelihood of a particular choice of letter chunking into English conversion units for an English word. Second approach considered contextual information of English and Japanese to calculate the plausibility of conversion using a single probability model. Last approach used probability models based on the *maximum entropy method* that can treat different kind information [19]. Results: *Corpus=15135, WA= 69.2%*

Nasreen Abdul Jaleel and Leah S Larkey(2003) developed a generative statistical model based on selected n-grams that produces a string of Arabic characters from a string of English characters. The model was set of conditional probability distributions over Arabic characters. Then it was conditioned on English unigrams and selected n-grams. Each English character n-gram e_i were mapped onto an Arabic character or sequence a_i with a probability $P(a_i|e_i)$. The model was trained from lists of proper name pairs in English and Arabic, via two alignment stages, the first of which was used to select n-grams for the model, and the second determined the translation probabilities for the n-grams. To generate Arabic transliterations for an English word, w_e, the word was first segmented according to the n-gram inventory. For each segment, all possible transliterations, w_a, were generated. For the alignment they used GIZA++[20].*Dataset and Results: Corpus=815, WA= 69.3% Top-1 and 71.2% for Baseline.*

Lee J and Chang S (2003) presented *statistical machine transliteration* approach in which source word to phonetic symbol conversion was not required. They demonstrated a framework to deal with the problem of acquiring English-Chinese bilingual transliterated word pairs from parallel-aligned texts. They used unsupervised learning approach in their system which automatically learns the parameters of the model from bilingual proper names. Along with the SMT, few hand crafted rules were also used both for translation and transliteration to improve the accuracy .The achieved excellent performance [21]. *Corpus Training=2430, WA = 86.0%, 94.4% Character Precision Rate and 96.3% Character Recall Rate.*

Li Haizhou, Zhang Min and Su Jian(2004) presented a method based on the *joint source channel model* for forward and backward transliteration. Their model simultaneously considered the source language and target language contexts in terms of n-grams (bigrams and trigrams) for machine transliteration. The key advantage was the use of bilingual contexts. The language pair used was *English-Chinese* . For this *English-Chinese* transliteration they used *noisy channel model* (NCM) and Bayes rule[22]. *Corpus =37,694, Word Error rates are presented.*

Malik M G A (2006) developed a *rule based* Punjabi Machine Transliteration (PMT) system that used rules for transliteration of *Shahmukhi* words into *Gurmukhi*. The PMT systems transliterate every word written in Shahmukhi into Gurmukhi. PMT was a special kind of machine transliteration. It converts a Shahmukhi word into a Gurmukhi word irrespective of the type constraints of the word. Their system preserved the phonetics of the transliterated word as well as the meaning [23]. *Dataset and Results: Corpus =45,420, WA=98.95%.*

Ekbal A et al.(2006)investigated a revised *joint source channel* based approach for *Bengali-English*. They used the regular expression to choose the transliteration units in the source word based on the inherent occurrences of consonants, vowels, and *matra*. Differing past and future contexts and context in the target word were examined. They used hand written transformation rules for 1:N alignments between English and Bengali in their system. In case of failure in alignment, even when incorporating handcrafted rules, manual intervention in the training phase was used to resolve the errors [24].*Dataset and Results: Corpus=6000, WA= 87.9%.*

Kumaran A and Kellner T (2007)developed a machine transliteration system based on the *noisy channel model.* In their frame work transliteration was obtained by calculating the parameters of the distribution that maximizes the likelihood of observing training data. Subsequently, given a target language string *t*, a posteriori was decoded the most probable source language string *s* that gave rise to *t*. The transliteration model $P(t|s)$ learned from the training corpus and $P(s)$ was the language model for the source language strings. The *Expectation Maximization (EM)* approach was used to exploit the information about the alignment, that some prefix (or suffix) of the source string must map to *some* prefix (or suffix, respectively) of the target string, in each of the paired

strings in the training set. They used Viterbi algorithm to find the optimal alignment. Language pairs used were *English to Hindi, Tamil, Japanese and Arabic* [25]. *Corpus Training=20,000, WA= 35.3% for Top-1 and 63.2% for Top-10 for exact match and 57.3% Top-1 and 89.8% for Top-10 for fuzzy match.*

Hermjakob U, Knight K and Daume H. (2008) developed a method to transliterate Arabic names into English. They used the SMT approach for transliteration. The system was trained on a bitext of 7 million sentences and Google's English terabyte n-grams and achieved better accuracy [26].*Dataset and Results: Corpus=1730,WA= 89.7%.*

Ganesh S et al.(2008)developed a SMT system which was language independent. Their developed the statistical model based on the *HMM alignment and CRF*. The HMM maximizes the probability word pairs using the EM algorithm. Then character level n-grams were set to maximum posterior predictions. This alignment was used to get character level alignment of the source and target language words. After the character level alignment, each source language character and its corresponding target language character were compared. CRF is used to generate a target language word from its source language word. CRF provided efficient training and decoding processes which was conditioned on both source and target languages. Their results showed that the hybridization of HMM and CRFperforms better. The language pair used was *English−Hindi* [27]. *Corpus=30,000 and 1000 out of Corpus, WAfor HMM = 69.3%, for HMM and CRF 72.1% for Top-5.*

Rama T and Gali K (2009) presented the transliteration for English-Hindi language pair using phrase based SMT technique. The major components of the system were GIZA++ and beam search based decoder. They varied the maximum phrase length from 2 to 7. The language model was trained using SRILM toolkit. They varied the order of language model from 2 to 8 [28].*Training and Development Data=9975, Testing=1000, WA=46.3%.*

Martin Jansche and Richard Sproat (2009) performed the named entity transcription with a pair of *n*-gram models at Google Inc. They used different size n–grams for different pairs. For *English-Korean*, a map was created between each Hangul glyph and its phonetic transcription in World-Bet based on the tables from Unitrans. The mapping between the Hangul syllables and their phonetic transcription was handled with a simple FST. The main transliteration model for the standard run was a 10-gram pair language model trained on an alignment of English letters to Korean phonemes. For the Indian languages *Hindi, Tamil and Kannada*, the same basic approach as for Korean was used. A reversible map was created between Devanagari, Tamil or Kannada symbols and their phonemic values, using a modified version of Unitrans. A 6-gram language model was used [29]. *Dataset and Results: Corpus=11,169 and additional Dictionary=9,047, WA=47.6%.*

Jong-Hoon Oh, KiyotakaUchimoto, and KentaroTorisaw's (2009)presented an approach which is based on two transliteration models – TM-G(**T**ransliteration **m**odel based on target language **G**raphemes) and TM-GP(**T**ransliteration **m**odel based on target language **G**raphemes and **P**honemes). The difference between the two models was whether or not a machine transliteration process depends on the target language phonemes. TM-G directly converts source language graphemes into target language graphemes, while TM-GP first transforms the source language graphemes into the target language phonemes and then the target language phonemes coupled with their corresponding source language graphemes were converted into the target language graphemes. They used three different machine learning algorithms - *CRF, a margin infused relaxed algorithm (MIRA),* and *MEM* for building multiple machine transliteration engines. The model was tested for *English to Chinese, Hindi, Tamil, Russian, Kannada,Japanese*

Katakana and *Korean Hangul* and from the *Japanese name to the Japanese Kanji language* pairs [30]. *Corpus=31,961, Training=2,896,WA=71.5% for English-Chinese for MEM-GP Model and 73.1 using Multiple Engine.*

SittichaiJiampojamarn et al.(2009) developed (*DIRECTL***)** an online discriminative sequence prediction model that employed an unsupervised many-to-many alignment using EM between the target and the source words. Their system have incorporated the three phases namely, input segmentation, target character prediction and sequence modelling. The feature vector consists of n-gram context features, HMM-like transition features, and linear-chain features. Finally, the most likely alignment for each word pair in the training data was computed with the standard Viterbi algorithm. The model was tested for *English to Chinese*, *Hindi*, *Russian*, *Japanese Katakana* and *Korean Hangul* and from the *Japanese name to the Japanese Kanji language* pairs [31]. *Corpus=31,961, Training=2,896, WA=74.6% E-C.*

Paul M, Finch A and Sumita E (2009) presented PB-SMT for Spanish-English language pair. The major components of the system were phrase-based SMT for character-level translation process, and a post-process filter to the SMT decoding process. Their experimentation showed that the incorporation of mixture models and phrase-based transliteration techniques largely out-performed standard phrase-based SMT engines gaining a total of 2.4% in BLEU and 2.1% in METEOR for the news domain [32]. *2.4% in BLEU and 2.1% in METEOR for the news domain*

Vijayanand k et al.(2009) developed a *rule based* transliteration system for *English to Tamil* by the partitioning algorithm and segmentation rules. The present system extracts the source names and stores them in an array list. These source names were retrieved from an array list sequentially and stored in a string variable for further processing. The value of the string was parsed character wise and then checked for the existence of a vowel or h, in the next two positions of its index i.e., for each character the next two characters were checked, if there exists a vowel or h, then these characters were extracted up to that index and stored in another string variable. Otherwise only that variable was stored and compared with the database that contains Tamil characters, for each combination of characters that are present in English. Thereafter each index in an array list of each transliteration was combined with each index in another array list of transliterated letter combination and then stored in another variable. This process continued until the system encounters the end of each array list [33]. *Corpus=1000, WA=40.39%*

Finch A and Sumita E (2009) developed a unified PB-SMT technique for English to eight multiple language pairs. Their technique did not consider language specific assumptions, dictionaries and phonetic information. The transliteration process directly transforms sequences of tokens in the source language into sequences of tokens in the target language. Multiple language pairs were transliterated by applying this technique in a single unified manner. The MT system was composed of two PB- SMT decoders. The first was generated from the first token of the target to the last one. The second system was generated the target from last to first one [34].*Corpus=31,961, Testing=2,896, WA=87.1% before and 90.8% after tuning.*

Xue Jiang et al.(2009) developed a *syllable based* name transliteration system to obtain the *Chinese* name from an *English* name. First, they syllabified the English name into a sequence of syllables by using handcrafted rules, and generated the most probable Pinyin sequence with the mapping model of English syllables to Pinyin (EP model), then converted the Pinyin sequence into a Chinese character sequence with the mapping model of Pinyin to characters (PC model).

The probability P of a transliteration from an English name to a Chinese name was denoted by P(Ch|En), the probability of a translation from an English syllable sequence to a Pinyin sequence was denoted by P(Py|En), and the probability of a translation from a Pinyin sequence to a

sequence of characters was denoted by P(Ch|Py).The character sequence in candidates having the max value of P(Ch|En) was the best transliteration [35].*Corpus=31,961, WA=49.8% for English-Chinese.*

Vijaya M S et al. K P (2009) presented a *rule based* transliteration system for English-Tamil language pair. They presented a transliteration model where the transliteration problem was modeled using classification technique. They used WEKA j48 decision tree classifier for implementation [36].*Corpus=6000, Testing=1000, WA=84.42% Top-1*

Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay(2009) presented three transliteration models for English to Hindi language pair. First model was *joint source-channel* model in which the context of previous and current translation unit was considered. In second model was the trigram model where the previous and the next source translation units were considered as the context. In third model, the previous and the next translation units in the source and the previous target translation units were considered as the context. This was the improved modified joint source-channel model.They also devised some post processing rules to remove the errors [37].*Corpus=9975, Testing=1000, WA=47.1% for Standard Run. Corpus=961,890 and WA= 38.9% Top-1 for Non-Standard Run.*

Chai Wutiwiwatchaiet al. (2010) developed a bidirectional *syllable based Thai-English* machine transliteration system [38]. This system relies on syllabification and the letter-to-sound mechanism. Thai-English was mostly done on the basis of sound mimicking of syllable units. The algorithm segments the input word in a source language into syllable like units and searches the pronunciations of each unit. The pronunciation units in the form of phonetic scripts were used to find possible transliteration forms given a syllable translation table. The best results were determined by using syllable n-gram. In the English to Thai system, a simple syllabification module of English words was created using the three steps. Step 1: Marking all vowels "a, e, i, o, u", e.g. - M[a]n[i]kr[a][o]. Step 2: Using some rules, merging consonantal letters surrounding each vowel to form basic syllables, e.g. Ma|ni|k|ra|o. Step 3: Post-processing by merging the syllable with "o" vowel into its preceding syllablee.g. Ma|ni|k|rao*Corpus=24,501, Testing=2000, WA=24.7% top-1 & Testing-1,994, WA= 9.3% Top-1 forbt.*

Josan G and Lehal G (2010) presented a *rule based* approach to improve Punjabi to Hindi transliteration. They used letter to letter mapping as the baseline transliteration and improved the accuracy by using rule based and Soundex based approaches.They have implemented and tested five different combinations for Punjabi-Hindi transliteration task [39]. *Corpus= Details not available,WA=92.65% for Base plus Rule plus Soundex approach.*

Manoj K. Chinnakotla, Om P. Damani and Avijit Satoskar (2010)proved that by using only the monolingual resourcesand handcrafted rules, it is possible to achieve reasonable transliteration performance. They achieved this performance by properly harnessing the power of *Character Sequence Modeling* (CSM), typically called the Language Model. Their system used CSM for word origin identification, character mapping rules to generate transliteration candidates, and then again CSM on the target side to rank the generated candidates. They have proved that if the word origin is used for the transliteration, then the system gives better results as compared to statistical methods [40]. *Corpus= 30,000, WA=75.1%*

HélaFehri, Kais Haddar and Abdelmajid Ben Hamadou(2011) developed a *rule based* method for recognition and transliteration for Arabic-French language pair related to sports venues names. They proposed an approach of recognition and translation based on a representation model of Arabic NEs and a set of transducers resolving morphological and syntactical phenomena. The representation model was based on the feature structure independent

of lexical categories. Their method integrated recognition and transliteration together using rule oriented approach. Implementation is done using the NooJ platform. They transliterated the proper names, the abbreviations and acronyms [41].*Corpus= 4000 Text From Sport Domain, Precision =981% , Recall=90% and F-measure=94%*

Deep K and Goyal V (2011) presented a transliteration method using a set of character mapping rules for Punjabi-English language pair. They addressed the problem of forward transliteration of person names. They used grapheme based method to model the transliteration problem. They demonstrated transliteration from Punjabi to English for conman names of persons, cities, states and rivers [42].*Dataset and Results: WA=93.22%*

Kaur J et al. (2011) presented a transliteration system which was developed by using SMT for English to Punjabi language pair. The major components of the system were MOSES for transliteration and set of rules for post processing [43].*Corpus=3844, Training=3200,Testing=644, WA=63.31%.*

Josan G and Kaur J (2011) presented a SMT based transliteration model (NCM) for transliterating the Punjabi text into Hindi text. They used two steps to obtain the transliteration. As a Baseline, they used a simple letter to letter based approach which maps Punjabi letters to the most likely letter in Hindi. Then a statistical model was developed and used for transliterating the Punjabi text into Hindi text [44].*Corpus=8000, Testing=1000, WA=87.72%.*

Dhore M L, Dixit S K and Sonwalkar T D (2012)presented machine transliteration of named entities for Hindi-English language pair using *CRF* as a statistical probability tool and n-gram as feature set. As the CRF calculates the probabilities over the entire input sequences, this approach was very good for the named entities of longer length. The results for tri-gram were expected more than the bi-gram as per the literature review carried out by them but it may not have happened due to the inadequacy of training data. They observed that CRF is well suited for the Indian languages, as most of the named entities are made up of multiple smaller named entities [45]**.***Corpus=7251, WA=85.79% for Bi-grams.*

Sharma S et al. (2012) presented a PB-SMT technique for English to Hindi transliteration. They used two different statistical applications MOSES and Stanford Phrasal for the transliteration. They performed four experiments using the combinations of two different notations UTF and wx with two different SMT applications namely Moses and Stanford Phrasal. They created the Language model using SRILM toolkit [46].*Corpus=20,000, Testing=400, WA=79.5% for Stanford Phrasal &wx format, WA=40.75% for Moses &wx format, WA=61.25% for Stanford Phrasal & UTF format, WA=31.75% for Moses & UTF.*

Kumar Pankaj and Kumar Vinod(2013) presented SMT based system to transliterate proper nouns written in Gurumukhi script of Punjabi language into English language. Their system first learns from the existing examples stored in the database and then uses n-gram approach to transliterate the new proper nouns of Gurumukhi Script into its equivalent English Language. In n-gram approach, they have used uni, bi, tri, four, five and six grams [47].*Corpus=15,000, Testing=2000, WA=97%*

Rathod PH, Dhore ML and Dhore RM (2013)developed a machine transliteration system for Hindi to English and Marathi to English language pairs using *Support Vector Machine* (SVM). They used phoneme and n-gram as features for their training. They used SVM as a machine learning algorithm for the classifications of patterns based on phoneme and variable n-gram sizes. In sequence labeling, they observed that as the n-gram size increases, it improves the accuracy. They observed that bi-gram gives good accuracy for the named entities having length two; tri-

gram gives good results length three. In their case, four-gram and five-gram accuracy was very close [48].*Corpus=10,000 Testing=5000, WA=86.52% for five-grams.*

BhallaDeepti et al. (2013) presented *rule based* transliteration system for English- Punjabi language pair. They used the syllabification approach. To convert English input to equivalent Punjabi output, they used NER Tool to first recognize the NEs from input sentence. The text entered by the user was first analyzed and then pre-processed. Then if the selected input was a proper name or location then it was passed to the syllabification module through which the syllables were extracted. After selecting the equivalent probability, syllables was combined to form the Punjabi word otherwise it was passed to the syllabification module and transliterated with the help of probability matching [49].*Corpus=25,500 Testing=6080, WA=88.19%*

Joshi H et al. (2013) have presented a transliteration system for Roman script to Devanagari script using *syllabification approach.* They retrieved the Hindi song lyrics written in the Roman script or Devanagari script. They used statistical machine learning approach for transliteration and TF-IDF model for information retrieval. Some rules werelso used for auto syllabification [50].*Corpus=22,500, Hindi song Lyrics=62,888, Queries Submitted=50, Correct Queries =25.*

6. REVIEW OF PHONEME BASED MODELLING

Arbabi M, Fischthal S M, Cheng V C and Bart E (1994)developed a hybrid algorithm to automate the transliteration process in real time using supervised neural networks and a knowledge based system to vowelize Arabic named entities. Short vowels are generally not written in Arabic script. A knowledge based system vowelize these names to add missing short vowels, and passes them to a neural network to determine whether they are reliable or unreliable in terms of Arabic syllabification. If reliable, then these names are converted to their phonetic representation using fixed transformation rules stored in a table. The phonetical representation is then transformed to the English script using another set of fixed rules. The transliteration model is therefore pre-defined in the form of fixed transformation rules. The main drawback of this study was that the importance of forming transformation rules is ignored. The emphasis was the vowelization of the names and separating Arabic and non-Arabic names through the syllabification process [51]. *Dataset and Results: Corpus=Phone Book Entries*

Knight K et al. (1998)modelled *Japanese to English* back-transliteration using the analogy of *source channel model*. They used only three steps in their implementation. Initially, a Japanese source word was transformed into its internal phonetical representation.Then, these source phonemes were mapped to their target English phonemes .Finally, phoneme to grapheme mapping used to generate the target English. They used weighted finite-state transducers (WFSTs) and a weighted finite-state acceptor (WFSA) for their implementation .They implemented P(w) in a weighted finite-state acceptor (WFSA) and implemented the other distributions in weighted finite-state transducers (WFSTs). P(w) denotes the generated written English word sequences. A WFSA is a state transition diagram which has weights and symbols on their transitionsin order to generate the output sequences .WFSA and WFSTs were built automatically as well as manually in the training stage, and transferred as a transliteration model. They implemented two algorithms for extraction, first were Dijkstra's shortest-path graph algorithm and second werek-shortest-paths algorithm [5]. *Corpus=100, WA=64%.*

Stephen Wan and Cornelia Maria Verspoor(1998)investigated a method to transliterate proper names from *English to Chinese* using phonetical representation. They introduce an algorithm for mapping from English names to Chinese characters based on heuristics about relationships between English spelling and pronunciation, and consistent relationships between English phonemes and Chinese characters. Their process consists of five main stages: Semantic

Abstraction, Syllabification, Sub-syllable Divisions, Mapping to Pinyin, and Mapping to Han Characters . A syllabification step segmented the English words to syllables, based on consonant boundaries. A sub-syllabification step further divided the syllables into sub-syllables to make them pronounceable within the Chinese phonemic set. In their case, the phoneme to grapheme transformation was based on a fixed set of handcrafted rules [52]. *Data is not available.*

Stalls B and Knight K (1998) presented a phoneme based back-transliteration for Arabic-English language pair. They build a model to transliterate names from Arabic into Roman Script. They build one new model, P (a|e), which converts English phoneme sequences directly into Arabic writing. They used a sequence of weighted finite-state transducers (WFSTs) for generating probable sequences and applied the EM learning algorithm described by Knight and Graehl in 1997 on their data [53].*Corpus=2800, Correct Transliterations=900*

Lee J S (1999)modelled *English to Korean* transliteration in two steps. The *English grapheme to English phoneme* transformation was modelled on the source channel model. The English phonemes were then transformed into Korean graphemes by using English to Korean standard conversion rules. These rules were in the form of context-sensitive rewrite rules[54]. *Corpus=1200, WA=47% Top-1 and 93% Top-10 using HMM and WA= 56% Top-1*

Jeong K S et al.(1999)developed a method of back-transliteration of words in*Korean* to*English*in the area of science and engineering which are not available in the dictionary.They did it by using two steps. In first stage, they detect an existence of foreign words by using statistical technique which relies on phonetic differences between English words and Korean words. In the second stage, they converted foreign words into their English origin words.The back-transliteration was carried out using a Hidden Markov Model (HMM), for which probabilities are calculated using training corpus.HMM based approach has been implemented by using Viterbi algorithm. For the string matching they used four well known algorithms namely, Damerau-Levenstein metric, n-gram, Soundex and Phonix algorithm [55]. *Corpus=1200, Testing=100, WA HMM=47% Top-1 and WA=56% Top-1for postprocessing.*

Jung S Y et al.(2000)modeled*English to Korean* transliteration with an extended Markov window, as the orthography of a language is strongly phonetic in the case of Korean. The method transforms an English word into English pronunciation by using a pronunciation dictionary. Then it segments the English phonemes into chunks of English phonemes; each chunk corresponds to a Korean grapheme as defined by handcrafted rules. Finally, their method automatically transforms the English phonemes chunk into Korean graphemes with the help of extended Markov window [56]. *Dataset and Results: Corpus=8368, WA=54.9% Top-10*

Meng H et al.(2001)proposed an *English to Chinese* transliteration method based on English grapheme to phoneme conversion, cross language phonological rules, rules for mapping between English and Chinese phonemes, as well as Chinese syllable based and character based language models. A set of hand-crafted transformations for locally editing the phonemic spelling of an English word to conform to rules of Mandarin syllabification are used to seed a transformation based learning algorithm. Their algorithm examines certain amount of data and learns the proper sequence of application of the transformations. Finally, it converts English phonemes to a Mandarin syllable sequence [57]. *Corpus: 3875, Testing: 1541, WA:47.1%*

Oh J H and Choi K S (2002) studied English-Korean transliteration model using pronunciation and contextual rules. They used phonetic information such as phoneme and its context. They also used word formation information such as English words of Greek origin. Pronunciations were taken from pronunciation dictionary to align the phonemes. Using the pronunciation of the

English word, a Korean word was generated [58]. *Dataset and Results: Corpus=6,185, Testing=1000, WA=67.83% and Character Accuracy=93.49%*

Lin W H and Chen H H(2002) presented a method of back-transliteration for *English to Chinese* language pair .They used a similarity based model for the task of backward transliteration, and provided a learning algorithm to automatically acquire phonetic similarities from a corpus. Their learning algorithm works on Widrow-Hoff rule with little modifications. According to their observations, learning algorithm converges quickly. Method using acquired phonetic similarities outperforms as compare to previous methods using pre-defined phonetic similarities or graphic similarities [59]. *Corpus=1574, Average Reciprocal Rank=83.22%*

Yan Q et al. (2003) presented a phoneme method for English to Japanese transliteration. They described a method for automatically creating and validating candidate Japanese transliterated terms of English words. A phonetic dictionary of English and a set of probabilistic mapping rules were used for generating transliteration candidates. A monolingual Japanese corpus was then used for automatically validating the transliterated terms. They evaluated the use of these extracted English and Japanese transliteration pairs with Japanese to English retrieval experiments using bilingual test collections [60].*Corpus=1469, WA=60% Top-, 82% Top-7*

Paola Virga and SanjeevKhudanpur (2003)developed the transliteration system for transliterating English names into Chinese to support of cross-lingual speech and text processing applications. They demonstrated the application of statistical machine translation techniques to "translate" the phonemic representation of an English name. It wasdone with the help of an automatic text to speech system. Then another statistical translation model is used to map the initial and final sequence to Chinese characters. The transliteration process is carried out using following steps. Firstly, they converted English name into a phonemic representation with the help of Festival speech synthesis system. Then, they translated English phoneme sequence into a sequence of Generalized Initials and Finals (GIF).It was followed by transformation of the GIF sequence into a sequence of pinyin symbols and finally the translation of the pinyin sequence to a character sequence. They used IBM SCM SMT [61].*Corpus=3835, Training=2233, Testing=1541, Mean Average Precision=0.517*

Gao W, Wong K F and Lam W (2004) investigated *English-Chinese* transliteration. They presented a statistical transliteration method for CLIR applications. An efficient algorithm for phoneme alignment was described. Unlike traditional rule-based approaches, their method was data-driven. They demonstrated comparable performance on accuracy to other systems [62-63].
Dataset and Results: Corpus=46,306, WA=36%, Character Accuracy=77%

Debasis Mandal D, Dandapat S, Gupta M, Banerjee P and Sarkar S (2007) developed a CLIR to retrieve English documents in response to queries written in Bengali and Hindi. They used dictionary based machine transliteration approach. The out-of-dictionary topic words were transliterated into English using a phonetic transliteration system. Their system works in the character level and converts every single Hindi or Bengali character in order to transliterate a word [64].*Dataset and Results: Mean Average Precision 78.95% and 36.49%, respectively*

Harshit Surana et al. (2008) proposed a Discerning Adaptable Transliteration Mechanism (DATM) method which applies different techniques based on the origin of the word. Their techniques also consider the properties of the source and target scripts. It does not need training data on the target side ; rather it uses more refined techniques on the source side . They used fuzzy string match method to compensate for lack of training on the target side. They have pointed out a variation in Indian words in Latin script and how to identify a word from the word origin. Two methods were suggested for the transliteration of Indian and foreign words

separately. Finally fuzzy string matching algorithm was used to generate the transliteration candidates. For calculating the distance between two letters they used Stepped Distance Function (SDF). Each letter was represented as a vector of features. Then, to calculate the distance between two strings, it used an adapted version of the Dynamic Time Warping algorithm. The language pairs used were *English-Hindi and English-Telugu* [65]. *Corpus=2000, MRR=0.44, Precision=45%*

SahaSujan et al. (2008)proposed a two-phase transliteration methodology. Transliteration module uses an intermediate alphabet to preserve the phonetic properties. They used gazetteer list as a dataset. English names in the name lists are transliterated to the intermediate alphabet. For the given English-Hindi word pair, if the transliterated intermediate alphabet strings were the same, then it was concluded that the English word was the transliteration of the Hindi word [66]. *Corpus=1070,WA=91.59*

Dhore M L, Dixit S K and Karande J B (2011) proposed a system which allows user to input the data in his local language/mother tongue or native language as well as to get the output reports in his his local language/mother tongue or native language .They discussed how to input data and get the information in Marathi, Hindi and Gujarati languages using transliteration approach based on the phonetic model.Their focus was to transliterate the input from multiple languages into common intermediate phonetic based code and to maintain the master database in English [67].*Corpus=3000, Results are available in terms of E_Score and L_Score*

DhoreML andDixit SK (2011)presented the use of transliteration approach for customizable localization support in small scale systems. They considered Marathi a Devanagari script based Indian language for the customizable localization support by using machine transliteration, translation memory and phonemic based pure consonant approach, sometimes also called as an half consonant approach. They demonstrated the support of local language access to the user to input and retrieve the data in Marathi on the fly with the help of middleware developed, whereas the data was stored in database in default language, English [68].*Corpus=1143, WA=95.97%*

Dhore M L, Dixit S K and Dhore R M (2012) focused on machine transliteration of Hindi to English and Marathi to English. They developed the transliteration tool using phonetic based direct approach. The tool does not require any training for bilingual database. They have shown in depth knowledge of word formation in Devanagari script based languages can provide better transliteration as compared to statistical approaches. Their model uses full consonant approach and metric based stress analysis for schwa deletion[69].

7. REVIEW OF HYBRID BASED MODELLING

Al-Onaizan Y and Knight K (2002)studied*Arabic-English* transliteration. They presented a transliteration algorithm based on sound and spelling mapping by using finite state machine. They combined phonetic based and grapheme based models into a single transliteration model. The transliteration score was calculated as a linear combination of phonetic based and grapheme based scores[70]. *Development Test Set=854 Blind Test Set=218, Word Accuracy=73%*

Bilac S and Tanaka H (2004) proposed a hybrid model based back-transliteration method for language pair *Arabic-English*. They improved the back-transliteration accuracy by combining the grapheme and pronunciation information [71]. *Dataset and Results: Corpus=714, WA=85% for Japanese-English (bt) and Corpus=150, WA=38% for Chinese-English (bt)*

Oh J H et al.(2005, 2006)investigated a hybrid method of spelling and phonetic based approaches for *English-Korean* and *English-Japanese* transliteration. They proposed a method for improving machine transliteration using the combination of three different transliteration models. As any one transliteration model alone has its own limitations on considering all possible transliteration behaviours, many transliteration models used in order to achieve a high-performance machine transliteration system. They described a method about transliteration production using the several machine transliteration models and transliteration ranking with web data and relevance scores given by each transliteration model [72-73].*Corpus=7172, WA=68% for English-Korean and Corpus=10,417, WA=62% for English-Japanese*

Malik Abbas et al. (2009)proposed a novel hybrid approach for Urdu to Hindi transliteration in which they combined FSM based techniques with statistical word language model. They filtered FSM output with the word language model to generate the correct Hindi output. They mainly dealt with the problem of removing the diacritical marks from the source input Urdu text. Their system produces the correct results in the form of diacritic marks is absent [74]. *Corpus=4,250 and CA=94.1%, 94.6%, 87.5%, 94.5% for four different hybrid models.*

Dhore M L et al(2012) discussed machine transliteration of names from Hindi and Marathi to English. The authors had taken a slightly different approach from the traditional statistical approaches using n-grams of the source and target names, by considering phonemes and word lengths as two main features for supervised learning. The approach proposed uses the source word (SW) is segmented into basic syllabic units and transliterated into English using full-consonant based mapping scheme. For this segmentation, two weights - based on diacritics used, and the length of source NE - were used, in order to do schwa deletion appropriately. The principled hybrid approach between linguistics (preparation of phonetic map, intermediate phonetic code) and statistical model (supervised learning of segmentation) was very appealing in this approach [75]. *Dataset and Results: Corpus=15,224 and WA=97.306%*

8. REVIEW OF COMBINED BASED MODELLING

Oh and Isahara (2007) studied *English-Korean* and *English-Japanese* transliteration using a combination of transliteration systems. They proposed a re-ranking method that makes use of confidence-score, language model, and Web-frequency features and combines them with machine-learning algorithms including SVM and MEM. Their testing of English to Korean and Japanese transliterations shown that individual transliteration models performed better in comparison to earlier approaches. The re-ranking algorithm had improved word accuracy [76].
Corpus=7172, WA=87.4% Top-1 for English-Japanese and Corpus=10,417, WA=87.5% for English-Korean for MEM, for SVM WA= 87.8% and WA=88.2%, respectively.

KarimiSarvnaz(2008) proposed a combined transliteration method using multiple grapheme based transliteration systems with the combination method being a mixture of a Bayes classifier and a majority voting scheme. They have explored many different approaches using n-grams, and proposed language-specific transliteration methods to improve transliteration accuracy. Their novel approaches use consonant-vowel sequences, and showed significant improvements over baseline systems. They also developed a new alignment algorithm, and examined novel techniques to combine systems.The system was evaluated for both *English-Persian* and *Persian-English* [77]. *Dataset and Results:Training and Testing =1500, WA=85.5% for English-Persian and Training and Testing =2010, WA=69.5% for Persian-English.*

9. OUTCOMES OF SURVEY

9.1 Common Features used for Machine Transliteration

Some of the commonly used features are Language Origin Detection, n-grams, Context window, Substrings, Prefixes,Suffixes, Orthographic, Phonetics, Phonemic, Syllabification, Root Information of the word, Stem Words, Syllables, Left and right context and Token length.

9.2 Classification of Machine Transliteration

Table 11. Classification of Machine Transliteration

Model	Learning Approach	
	Rule Based Learning	**Statistical Based Learning**
Grapheme Model	Language Model, Direct Example Based, Character Sequence, Modelling, Syllable Based ModelLetter To Phoneme Model(L2M)	SMT, Noisy Channel Model, Source Channel Model, Joint Source Channel Model, N-gram Model,Hidden Markov Model, Maximum Entropy, Conditional Random Fields, ,Decision Trees, Support Vector Machine
Phoneme Model		Weighted Finite State Transducers Markov Window(MW) Transformation Based Learning Model
Hybrid Model	HMM and Rule Based , CRF and Rule Based etc	
Combined Model	Multiple Phoneme Based or Multiple Grapheme Based methods	

9.3 Pros and Cons of Machine Transliteration Models

Grapheme Model: Advantages: Less number of steps required. Less error propagation. Fewer linguistic resources required. Performs better than or at par with phoneme based approaches. Language independent.Well suited for statistical probability. **Disadvantages:** Large amount of corpus and training required for reasonable accuracy.

Phoneme Model: Advantages: Elevating the role of pronunciation. If mapping table used, no training data required.**Disadvantages:** Multiple steps required in process. Error Propagation due to multiple steps.Rely on bilingual pronunciations resources. In most cases it is Language dependent. Not well suited for statistical probability.

Hybrid Model: Advantages: Better Results.Initial results are promising.**Disadvantages:** Implementation Complexity.

Combined Model: Advantages: Better Results.Initial results are promising.**Disadvantages:**Implementation Complexity.

9.4 Pros and Cons of Machine Learning Models

Linguistic Approach: Advantages: Easy to implement Can give better result than statistical methods by enriching language specific rules. Provide good performance at a relatively high system engineering cost. **Disadvantages:** Huge experience and grammatical knowledge of

particular language is required. Not transferable to other languages. It is not trainable. Language dependent.High engineering cost.

Statistical Approach: Advantages: It is trainable. It is adaptable. It is scalable. Low maintenance.It is Language independent approach. **Disadvantages:** Sufficient training data is required to achieve good result. Corpora are not available for most of the languages.

9.5 Summary

From the above survey, it is clear that the three approaches are most popular for machine transliteration. One of these is Statistical Machine Transliteration (SMT), second of these is Conditional Random Fields (CRF) and last one is Hidden Markov Model (HMM). For grapheme based statistical methods, parallel corpus is required and need to be trained for the adequate number of entries using one of the learning approaches such as HMM, CRF, SVM etc. One of the concrete observations is that, better accuracy can be achieved only by using Hybrid and Combined models. As each individual model has its own limitations, other model can be used in combination to overcome those limitations. Initial results of Hybrid and Combined models are promising and need to be used for further research work in this area.

REFERENCES

[1] Karimi S, Scholer F, & Turpin, (2011) "Machine Transliteration Survey", ACM Computing Surveys, Vol. 43, No. 3, Article 17, pp.1-46.
[2] Antony P J &Soman K P, (2011) "Machine Transliteration for Indian Languages: A Literature Survey", International Journal of Scientific and Engineering Research, Vol 2, Issue 12, pp. 1-8.
[3] Jong-Hoon Oh, Key-Sun Choi & Hitoshi Isahara, (2006) "A Comparison of Different Machine Transliteration Models", Journal of Artificial Intelligence Research, pp. 119-151.
[4] Brown P F, Pietra V J D, Pietra S A D, & Mercer R L, (1993) "The Mathematics of Statistical Machine Translation: Parameter estimation", Computational Linguistic, 19, 2 pp. 263–311.
[5] Knight Kevin &Graehl Jonathan, (1998) "Machine Transliteration", In Proceedings of the 35th Annual Meetings of The Association for Computational Linguistics, pp. 128-135..
[6] Li Haizhou et al., (2004) "A Joint Source-Channel Model for Machine Transliteration", ACL.
[7] L Rabiner, (1989) "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of IEEE, Vol. 77, No. 2, pp. 257-296.
[8] Phil Blunsom, (2004) "Hidden Markov Models".
[9] J Lafferty et al., (2001) "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", In International Conference on Machine Learning.
[10] Hanna M. Wallach, (2004) "Conditional Random Fields: An Introduction", University of Pennsylvania CIS Technical Report MS-CIS-04-21.
[11] Charles S et al., "An Introduction to CRF Relational Learning", University of Massachusetts.
[12] A L Berger, S D Pietra, & V J Della Pietra, (1996) "A Maximum Entropy Approach to Natural Language Processing", Computational Linguistics, vol. 22, no. 1, pp. 39–71.
[13] K.P.Soman et al, Machine Learning with SVM and Other Kernel Methods, Book, PHI.
[14] Y. Yuan et al. (1995) Fuzzy sets and Systems, pp 125-139.
[15] Lee J S & Choi K S, (1998) "English to Korean Statistical Transliteration For Information Retrieval", Computer Processing of Oriental Languages.
[16] Kang I H et al., (2000) "English-to-Korean Transliteration Using Multiple Unbounded Overlapping Phoneme Chunks", In Proceedings of the 18th Conference on Coling, pp. 418–424.
[17] Kang B J et al., (2000) "Automatic Transliteration & Back-Transliteration by Decision Tree Learning", 2nd International Conference on Language Resources and Evaluation.
[18] Kang B J (2001) "A Resolution of Word Mismatch Problem Caused by Foreign Word Transliterations and English Words in Korean Information Retrieval", Ph.D. Thesis, KAIST.
[19] Goto I et al, (2003) "Transliteration Considering Context Information Based on the Maximum Entropy Method", In Proceedings of MT-Summit IX, pp. 125-132.

[20] Jaleel et al, (2003) "Statistical Transliteration For English-Arabic Cross Language Information Retrieval", 12th International Conference on Information and Knowledge Management.
[21] Lee, J et al., (2003), "Acquisition of English-Chinese Transliterated Word Pairs from Parallel-Aligned Texts using a Statistical Machine Transliteration Model", HLT-NAACL 2003.
[22] Li H et al., (2004) "A Joint Source-Channel Model for Machine Transliteration", ACL.
[23] Malik M G A, (2006) "Punjabi Machine Transliteration", In Proceedings of the 21st International Conference on Computational Linguistics, ACL, pp.1137-1144.
[24] Ekbal A, Naskar S &Bandyopadhyay S, (2006) "A Modified Joint Source Channel Model for Transliteration", In Proceedings of the COLING-ACL, Australia, pp.191-198.
[25] Kumaran A et al., (2007) "A Generic Framework for Machine Transliteration", 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.
[26] Hermjakob, U. Et al., (2008) "Name Translation in Statistical Machine Translation Learning When to Transliterate", Proceedings of Association for Computational Linguistics, pp. 389–397.
[27]Ganesh S, Harsha S, Pingali P, &Verma V, (2008) "Statistical Transliteration for Cross Language Information Retrieval Using HMM Alignment and CRF", In Proceedings of the Workshop on CLIA, Addressing the Needs of Multilingual Societies.
[28] Rama T. Et al., (2009) "Modeling Machine Transliteration as a Phrase Based Statistical Machine Translation Problem", Proceedings of the 2009 Named Entities Workshop, pp. 124-127.
[29] Martin Jansche& Richard Sproat, (2009) "Named Entity Transcription with Pair n-Gram Models", Google Inc., Proceedings of the 2009 Named Entities Workshop, Singapore pp. 32–35.
[30] Jong-Hoon Oh et al., (2009) "Ma-chine Transliteration Using Target-Language Grapheme and Phoneme: Multi-Engine Transliteration Approach", Named Entities Workshop, pp. 36–39.
[31] SittichaiJiampojamarn et al, (2009) "DirecTL: a Language Independent Approach to Transliteration", Proceedings of the 2009 Named Entities Workshop, Singapore, pp. 28–31.
[32] Paul, M. Et al., (2009) "Model Adaptation and Transliteration for Spanish-English SMT", Proceedings of the 4th EACL Workshop on Statistical Machine Translation, pp. 105-109.
[33] KommaluriVijayanand, (2009) "Testing and Performance Evaluation of Machine Transliteration System for Tamil Language", Proceedings of the 2009 NEWS, pp. 48–51.
[34] Finch, A. &Sumita, E, (2009) "Transliteration by Bidirectional Statistical Machine Translation", Proceedings of the 2009 Named Entities Workshop, pp. 52-56.
[35] Xue Jiang, Le Sun &Dakun Zhang, (2009) "A Syllable-Based Name Transliteration System", Proceedings of the 2009 Named Entities Workshop, Singapore, pp. 96–99.
[36] Vijaya M.S. et al., (2009) "English to Tamil Transliteration using WEKA", International Journal of Recent Trends in Engineering, Vol. 1, No. 1, pp. 498-500.
[37] Das A., Ekbal A., Mandal T. &Bandyopadhyay S, (2009) "English to Hindi Machine Transliteration System at NEWS", Proceedings of the 2009 Named Entities Workshop pp.80-83.
[38] Chai Wutiwiwatchai and AusdangThangthai, (2010) "Syllable-based Thai-English Machine Transliteration", Named Entities Workshop Sweden pp. 66-70.
[39] Josan, G. &Lehal, G, (2010) "A Punjabi to Hindi Machine Transliteration System", Computational Linguistics and Chinese Language Processing, Vol. 15, No. 2, pp. 77-102, 2010.
[40] Chinnakotla M K, Damani O P, and Satoskar A, (2010) "Transliteration for Resource-Scarce Languages", ACM Transactions on Asian Language Information Processing, 9, 4, pp. 1-30.
[41] Fehri H et al., (2011) "Recognition and Translation of Arabic Named Entities with NooJ Using a New Representation Model", 9th International Workshop on FSM and NLP, pp.134–142.
[42] Deep, K. &Goyal, V, (2011) "Development of a Punjabi to English Transliteration System", International Journal of Computer Science and Communication, Vol. 2, No. 2, pp. 521-526.
[43] Kaur, J. &Josan, G, (2011) "Statistical Approach to Transliteration from English to Punjabi", International Journal on Computer Science and Engineering, Vol. 3, No. 4, pp. 1518-1527.
[44] Josan, G. &Kaur, J, (2011) "Punjabi To Hindi Statistical Machine Transliteration", International Journal of Information Technology and Knowledge Management, pp. 459-463.
[45] Dhore Manikrao L, Dixit Shantanu K and Sonwalkar Tushar D, (2012) "Hindi to English Machine Transliteration of Named Entities using Conditional Random Fields", International Journal of Computer Applications, Vol. 48– No.23, pp. 31-37.
[46] Sharma S. Et al., (2012) "English-Hindi Transliteration using Statistical Machine Translation in different Notation", International Conference on Computing and Control Engineering.

[47] Kumar, P. and Kumar, V, (2013) "Statistical Machine Translation Based Punjabi to English Transliteration System for Proper Nouns", International Journal of Application or Innovation in Engineering & Management, Vol. 2, Issue 8, pp. 318-321.
[48] Rathod P H, Dhore M L and Dhore R M, (2013) "Hindi And Marathi To English Machine Transliteration Using SVM", International Journal on Natural Language Computing (IJNLC) Vol. 2, No.4, pp. 55-71.
[49] Bhalla, D. and Joshi, N, (2013) "Rule Based Transliteration Scheme For English To Punjabi", International Journal on Natural Language Computing, Vol. 2, No. 2, pp. 67-73.
[50] Joshi, H., Bhatt, A. & Patel. H, (2013) "Transliterated Search using Syllabification Approach", Forum for Information Retrieval Evaluation.
[51] Arbabi M, Fischthal S M, Cheng V C & Bart E, (1994) "Algorithms for Arabic Name Transliteration", IBM Journal of Research and Development, pp. 183-194.
[52] Stephen Wan & Cornelia Maria Verspoor, (1998) "Automatic English-Chinese Name Transliteration for Development of Multilingual Resources", NSW 2109, pp. 1352-1356.
[53] Stalls, B. & Knight K, (1998) "Translating Names and Technical Terms in Arabic Text", COLING ACL Workshop on Computational Approaches to Semitic Languages, pp. 34-41, 1998.
[54] Lee J S, (1999) "An English-Korean Transliteration and Re-transliteration Model for Cross-Lingual Information Retrieval", Computer Science Dept., KAIST.
[55] Jeong K S et al., (1999) "Automatic Identification and Back-Transliteration of Foreign Words for Information Retrieval", Information Processing and Management, 35, 4, pp. 523–540.
[56] Jung S Y et al., (2000) "An English to Korean Transliteration Model of Extended Markov Window", In Proceedings of the 18th Conference on Computational linguistics, pp. 383–389.
[57] Meng H et al., (2001) "Generating Phonetic Cognates to Handle Named Entities in English-Chinese Cross-Language Spoken Document Retrieval", ASRU '01, pp. 311-314.
[58] Oh J H, & Choi K S, (2002) "An English-Korean Transliteration Model using Pronunciation and Contextual Rules", In Proceedings of COLING 2002, pp. 758-764.
[59] Lin W H & Chen H H, (2002) "Backward Machine Transliteration by Learning Phonetic Similarity", In Proceedings of the 6th Conference on Natural Language Learning, pp. 1–7.
[60] Yan, Q et al., (2003) "Automatic Transliteration For Japanese-to-English Text Retrieval", ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 353-360.
[61] Paola Virga et al.,(2003) "Transliteration of Proper Names in Cross-Lingual Information Retrieval", Proceedings of the ACL Workshop on Multilingual and Mixed-language NER.
[62] Gao W, Wong K F, & Lam W, (2004) "Improving Transliteration with Precise Alignment of Phoneme Chunks and Using Contextual Features", vol. 3411, Springer, Berlin, pp. 106–117.
[63] Gao W, Wong K F, & Lam W, (2004) "Phoneme-based Transliteration of Foreign Names for OOV Problem", First IJCNLP, vol. 3248, Springer, pp. 110–119.
[64] DebasisMandal, D., Dandapat, S., Gupta, M., Banerjee, P. &Sarkar, S, (2007) "Bengali and Hindi to English CLIR Evaluation", Cross-Language Evaluation Forum CLEF, pp. 95-102.
[65] HarshitSurana& Anil Kumar Singh, (2008) "A More Discerning and Adaptable Multilingual Transliteration Mechanism for Indian Languages", Proceedings of the Third IJCNLP, pp. 64-71.
[66] Saha S et al., (2008) "NE Recognition in Hindi Using Maximum Entropy and Transliteration".
[67] M L Dhore, S K Dixit and J B Karande, (2011) "Cross Language Representation for Commercial Web Applications in Context of Indian Languages using Phonetic model", CiiT International Journal of Artificial Intelligent Systems and Machine Learning, Volume 3, No. 4, pp 174-179.
[68] M L Dhore and S K Dixit, (2011) "Development of Bilingual Application Using Machine Transliteration: A Practical Case Study", CiiT International Journal of Artificial Intelligent Systems and Machine Learning, Volume 3, No. 13, pp 859-864.
[69] M L Dhore, S K Dixit and R M Dhore, (2012) "Hindi and Marathi to English NE Transliteration Tool using Phonology and Stress Analysis", 24th International Conference on Computational Linguistics, Proceedings of COLING: Demonstration Papers, at III, Bombay, pp 111-118.
[70] Al-Onaizan& Knight K, (2002) "Machine Transliteration of Names in Arabic Text", Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages.
[71] Bilac S, & Tanaka H, (2004) "Improving Back-Transliteration by Combining Information Sources", In Proceedings of IJCNLP2004, pp. 542-547.
[72] Oh J H & Choi K S, (2005) "Machine Learning Based English-to-Korean Transliteration using Grapheme and Phoneme Information", IEICE Transaction on Information and Systems.

[73] Oh J H & Choi K S, (2006) "An Ensemble of Transliteration Models for Information Retrieval", Information Processing and Management, 42, 4, pp. 980–1002.
[74] Abbas Malik et al, (2009) "A Hybrid Model for Urdu Hindi Transliteration", Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 177–185.
[75] M L Dhore, S K Dixit and R M Dhore, (2012) "Optimizing Transliteration for Hindi/Marathi to English Using only Two Weights", Proceedings of the First International Workshop on Optimization Techniques for Human Language Technology, COLING,IITB, pp 31–48,
[76] Oh J H and Ishara H, (2007) "Machine Transliteration using Multiple Transliteration Engines and Hypothesis Re-Ranking", In Proceedings of the 11th Machine Translation Summit.
[77] SarvnazKarimi, (2008) "Machine Transliteration of Proper Names between English and Persian", Thesis, RMIT University, Melbourne, Victoria, Australia.