Machine transliteration: A Review of Literature

Kanwaljit Kaur
Department of Computer Science
Punjabi University Patiala, Punjab, India

Abstract- Machine transliteration is an emerging research area which converts words from one language to another without losing its phonological characteristics. Transliteration is a supporting tool for machine translation and Cross language information retrieval. Transliteration is mainly used for handling named entities and out of vocabulary words in a machine translation system. It preserves the phonetic structure of the words. This paper discusses the various challenges, approaches and existing systems in transliteration. The major challenges in developing a transliteration system are missing sounds, zero or multiple character mappings, differences between scripts etc. The approaches for the transliteration system can be phoneme based, grapheme based or combination of both. Few researches that have taken place in the field of transliteration are listed in this paper, although the list may not be exhaustive.

Index Terms- Transliteration, Machine translation, Cross Language Information Retrieval, Named Entities.

I. INTRODUCTION

Transliteration converts the text from one script to another. Systematic transliteration refers to the conversion of a word in source script to a word in target script such that the target language word is:

- Phonetically equivalent to the source language word e.g. Mumbai → ਮੁੰਬਈ
- Conforms to the phonology of target language e.g. Narendar → ਨਰੇਂਦਰ
- Matched with the source language word by considering the orthographic character usage in target language.

Transliteration [1] can be seen as two level processes: first segmenting the source language word into transliteration units and then aligning and mapping these units to target language units.
For e.g. the word "Mera" which can be segmented as 'm', 'e', 'r', 'a' , then these units are transliterated to target language units as ' ਮ', 'ੇ' , ' ਰ ', 'ਾ'. Transliteration is mainly used to convert the foreign words in a language
which are required to be phonetically but need not to be grammatically equivalent to the words in another language.
Transliteration may define complex conventions and tries to be more perfect to enable the reader to recalculate the spellings of the original words. Thus, transliteration should preserve the syllable sounds in the words. Transliteration can be of two types namely forward and backward transliteration. Transliteration of a word from its native script to foreign script is called forward transliteration. Restoring previously transliterated word to its native script is called backward transliteration.

Machine translation decodes the meaning of the source text and re-encode the meaning in target language using various approaches as dictionary translation and statistical or example based translation. But when many crucial issues like out of vocabulary words, Proper nouns and other technical terms needs to be handled, transliteration approaches are taken to solve these issues. Thus machine transliteration usually supports machine translation and helps preventing translation errors when translations of proper names and technical terms do not exist in translation dictionary. The general transliteration model consists of two stages: Training running on a bilingual corpus and Transliteration. Training stage comprises of aligning the source-target words at character or sound level and rule generation. The transliteration stage segments the new (test) source word and generates appropriate transliteration.
In this survey paper, we are discussing about some of the challenges that a transliteration system may face including script differences, missing sounds, language of origin etc. The section 3 lists the various approaches and existing transliteration systems.

II. COMMON CHALLENGES IN TRANSLITERATION

A. ***Script Differentiation:-*** The main hurdle transliteration system needs to tackle is the difference between source and target language script. A script represents text using set of useful symbols. Script represents one or more writing systems. For example Devnagri is the script for over 120 languages including Hindi, Nepali, Sindhi; Maithili etc. Thus one script can be used for multiple languages. On the other hand, one language can be written in multiple scripts as Japanese can be written in Hiragana, Katakana and kanji ideographs. Another important issue is the direction in which a script is written. The language like Persian, Arabic are written from Right To Left (RTL) whereas the English and other languages are written form Left to Right (LTR).

B. ***Missing Sounds:-*** All the languages have their own phonetic structure, and symbols. If there is a missing phonetic in the letters of a language, then those phonetic are represented using digraphs and tri-graphs. Transliteration systems needs to take care of the convention of writing the missing phonetics in each of the languages involved in transliteration.

C. ***Multiple Transliterations:-*** Based on the opinion of different humans, a source term can have multiple valid transliterations. Different dialects in the same language can also lead to transliteration variants. Multiple transliterations certainly affect the accuracy of a system as gathering all possible variants of a word in a corpus is not feasible.

D. ***Language Of Origin:-*** Named entities can have multiple transliterations and each transliteration is correct according to the context under consideration. So, these words can be sometimes transliterated by considering local context and sometimes considering global context. One challenge would be which letters to choose to represent the origin of the word. The name Razaq has the Arabic origin while it is written as Razak in Indian origin [2].

E. ***Transliterate Or Not:-*** Whether a word should be translate or transliterate, deciding this phenomena is a big challenge. Place names and organization names are the most common cases where both translation and transliteration are necessary. For example, the word "Kashmir Valley" needs both translation and transliteration.

III. MACHINE TRANSLITERATION APPROACHES

Many different transliteration methods have been proposed in literature leading to the variations in methodologies and language supported. Due to many different variations categorization of transliteration approaches is not very straightforward. One categorization possible is based on information sources used in the process. The categorization is as follows:

- Grapheme based approaches that consider transliteration as orthographic process and use spellings.
- Phoneme based approaches consider the task as purely phonetical process and use phonetics.
- Hybrid approach that mixes up the above two approaches.
-

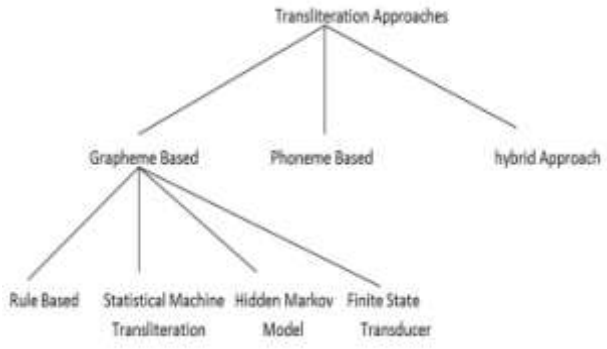

Fig. 1 Classification of transliteration approaches

A. ***Grapheme Based Models:-*** Grapheme based transliteration [1] is a process of mapping a grapheme sequence from a source language to a target language ignoring the phoneme level processes. In this approach characters from source language are directly mapped to characters of target language. So, they are also called direct methods. This approach mainly relies upon statistical information that can be obtained from characters. Grapheme based models are classified into the Statistical

Machine Transliteration (SMT) based model, Rule based models, Hidden Markov Model , Finite State Transducer (FST) based model.

B. ***Rule Based Approach:-*** In Rule based approach, set of rules are specified by human experts in order to map a source sentence segment to representation in target language sentence. Rules are generally based on the morphological, syntactic and semantic information of the source and target languages. Rules are very important for various stages of translation such as syntactic processing, semantic interpretation and contextual processing of the language. Transliteration in rule based system is done by pattern matching of the rules. The success lies in avoiding the pattern matching of unfruitful rules. General world knowledge is required for solving interpretation problems such as disambiguation.
Ali and Ijaz(2010) have developed "English to Urdu transliteration system which is based on rule based approach. Kak et al. (2010) have developed a rule based converter for Kashmiri language for Persio-Arabic script to Devnagari script.

C. ***SMT Approach:-*** Statistical approach [3] tends to be easier than generating handcrafted rules. In this approach, translations are based on mathematical model whose parameters are derived from the analysis of bilingual text corpora. Every sentence in the target language is the translation of the source language sentence with some probability. The sentence having highest probability is the required translation. This approach finds the most probable English sentence given a foreign language sentence and automatically aligns the words within sentences in the parallel corpus, then probabilities are determined automatically by training statistical model using parallel corpus. So, sentences get transliterated based on the probabilities. The SMT approach is more advantageous than rule based approach as it efficiently uses human and data resources. There are many parallel and monolingual corpora available in machine readable format. Generally SMT systems are not tailored to any specific pair of languages. Moreover rule based systems require rules to be made manually which is very costly and time consuming. Lee and Chang(2003) have developed an English Chinese transliteration system based on Statistical Model. Malik(2013) has developed a system for transliterating Urdu to Hindi based on statistical approach.

D. ***FST approach:-*** Finite State Transducers [3] are being used in different areas of pattern recognition and computational linguistics. A finite state transducer is a finite state machine having an input and output tape and has an intrinsic power of transducing or transliterating. When transducer shifts from one state to another, it will print a word as an output. So transducer can accept the word in one language and can produce transliteration in another language. So, transducer can be seen as a bilingual generator. It is a network of states which are labeled with input and output symbols and transition between them. Starting from initial state and walking through the end state, FST can transform an input string by matching it with input labels and produce a corresponding output string using output labels.
Knight and Graehl(1998) have developed a phoneme based back transliteration model from Japanese to English using Finite State Transducer.

E. ***HMM (Hidden Markov Model) Approach:-*** Hidden Markov Model is a statistical model in which the system is assumed to have hidden states. The model has a set of states each having a probability distribution. Transitions between the states are controlled by set of probabilities called transition probabilities. In HMM, the state is not visible but output dependent upon the state is visible. The translation is achieved according to the associated probability at a particular state.

F. ***Phoneme Based Model:-*** Phonemes are the smallest significant units of sound. In phoneme based approach, the written word of source language is mapped to written word of target language via the spoken form associated with the word. Phoneme based method [1] [3] is also known as Pivot method. The reason for using this approach is that phonetical

representation makes it possible to use it as an intermediate form between source and target languages (Similar to Interlingua MT). The other reason for the interest in phonetic based transliteration is its ability to capture the pronunciation of the words. 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. Phonetic-based methods identify phonemes in the source word W, produce source language phonemes (P) and then map the phonetical representation of those phonemes (P) to character representations in the target language to generate the target word(s) T.

In phoneme based approaches, the transliteration key is the pronunciation of the source phoneme rather than spelling or the source grapheme. The phoneme based approach has also received remarkable attention in various works. Based on phonology, the source text can be transliterated to target text in terms of pronunciation similarities between them. The syllables are mapped to phonemes, based on some transcription rules. The mapping templates between phonemes of source and target language are the transliteration rules.

G. ***Hybrid and Correspondence based Models:-*** The Correspondence and hybrid [1] transliteration model makes use of both source language graphemes and source language phonemes when producing target language transliterations. Both models can be combination of two or more transliteration approaches. These can be combination of grapheme and phoneme based models or combination of two grapheme models for e.g. Rule based and statistical. The correspondence based model makes use of the correspondence between a source grapheme and a source phoneme when it produces target language graphemes; the hybrid model simply combines grapheme and phoneme through linear interpolation.

Some examples of Hybrid models are:

- Grapheme Based + Phoneme Based
- Rule Based + SMT

H. **Conditional Random Field: -** It is a class of statistical modeling techniques often applicable in machine learning and pattern recognition. CRF [4] is used for structured prediction by labeling sequential data such as natural language text. CRF predicts a label for a single sample by taking context into account i.e. by considering neighboring samples. In CRF, each feature function takes a sentence s, the position I of a word in the sentence, label l_i of the current word and label l_{i-1} of the previous word. It outputs a real-valued number. Each feature is assigned a weight and finally, these are transformed into probabilities. Usually gradient decent method is used for training the CRF model.

I. **Support Vector Machine: -** Support vector machines [5] are supervised learning models associated with learning algorithms that analyze data used for classification. SVM training algorithm builds a model that divides training data into number of categories and assigns new examples to one of those categories. An SVM model is a representation of examples as points in space, mapped so that the examples are divided into different categories with a clear gap. New examples are made to belong to a category based on which side of the gap they fall on.

Supervised learning is not possible in case of unlabeled data, so an unsupervised learning clusters the data into groups, and map new data to these formed groups. The clustering algorithm is called support vector clustering and is often used in industrial applications either when data is not labeled or when only some data is labeled as a preprocessing for a classification pass.

J. **Decision tree learning: -** Decision tree [6] is a predictive model which maps item observations to conclusions about the item's target value. Tree models where the target variable can take a finite set of values are called classification trees. In these tree structures, class labels are represented by the leaves and branches represent conjunctions of features that lead to those class labels. In decision analysis, a decision tree can be used to represent decisions and decision making.

The goal of decision tree learning is to create a model that predicts the value of a target variable based on several input variables.

K. Neural Network: - neural network is trained to maximize translation performance. It is a radical departure from the phrase-based statistical translation approaches, in which a translation system consists of subcomponents that are separately optimized. The artificial neural network (ANN) [7] is a unique learning algorithm based on the working and structure of human brain. With use of ANN, it is possible to perform a number of tasks, such as classification, clustering, and prediction, using machine learning techniques like supervised or reinforced learning. A bidirectional recurrent neural network (RNN), also called *encoder*, encodes source sentence for a second RNN, known as a *decoder* that is used to predict words in the target language.

Machine translation or transliteration models are inspired by deep representation learning. Their memory requirements are very less as compared to statistical models. Neural Networks are trained after for certain domains or applications. After training, the network practices. With time it starts operating according to its own judgment, turning into an "expert".

IV. LITERATURE SURVEY

Arbabi et al. developed an Arabic-English transliteration system [2] using knowledge-based systems and neural networks. The first step in this system was to enter the names into the database which was obtained from telephone dictionary. As in Arabic script, short vowels are generally not written, a knowledge-based system is used to vowelized these names to add missing short vowels. The KBS system accepts all unvowelized names and generates all possible vowelizations conforming to Arabic name. The words which cannot be properly vowelized by KBS are then eliminated using artificial neural network. The network is trained using cascade correlation method, a supervised, feed forward neural processing algorithm. Thus the reliability of the names in terms of Arabic syllabification is determined through neural networks. The output of the network is in binary terms. If the node fires with a threshold of 0.5, then the word is given to KBS for vowelization otherwise set aside to be vowelized in some other way. The artificial neural network is trained on 2800 Arabic words and tested on 1350 words. After this, the vowelized names are converted into phonetic roman representation using a parser and broken down into groups of syllables. Finally the syllabified phonetics is used to produce various spellings in English. KBS vowelize almost 80% of the names but with higher percentage of extra vowelizations while ANN vowelizes over 45% of the names with very low rate of errors.

Wan and Verspoor have proposed an "Automatic English-Chinese name Transliteration" [8] system. The system transliterated on the basis of pronunciation. That is, the written English word was mapped to written Chinese character via spoken form associated with the word. The system worked by mapping an English word to a phonemic representation and then mapping each phoneme to a corresponding Chinese character. Since the phoneme-to-grapheme process is considered the most problematic and least accurate step, they limited their model to place names only. The transliteration process consisted of five stages: Semantic Abstraction, Syllabification, Sub-syllable divisions, Mapping to Pinyin and Mapping to Han characters. Semantic abstraction was a preprocessing step that performed dictionary look-ups to determine which parts of the word should be translated or which should be transliterated. As Chinese characters are monosyllabic so each word to be transliterated was divided into syllables. The outcome of the syllabification process was a list of syllables each with at least one vowel part. A sub-syllabification step further divided the syllables into sub syllables to make them pronounceable within the Chinese phonemic set. The phonetic representation of each sub syllable was transformed to Pinyin, which is the most common standard Mandarin Romanization system. Another fixed set of rules transforms Pinyin to Han (Chinese script). Therefore, the transliteration models were divided into a grapheme-to-phoneme step and a phoneme-to-grapheme transformation which was based on a fixed set of rules.

Kang et. al. presented an English-to-Korean automatic transliteration and back transliteration system [9] based on decision tree learning. The proposed methodology is fully bidirectional. They have developed very efficient character alignment

algorithm that phonetically aligns the English words and Korean transliteration pairs. The alignment reduces the number of decision trees to be learned to 26 for English-to-Korean transliteration and to 46 for Korean-to-English back transliteration. After learning, the transliteration and back transliteration using decision tree is straightforward.

Oh et. al. have developed an "English to Korean Transliteration System based on correspondence model [10] by using both phonetic information and Orthography. This system first performs alignment and then transliteration. The proposed system is composed of two main parts: data preparation and machine transliteration. The data preparation step creates training data by devising an EPK alignment algorithm. The EPK alignment algorithm recognizes the correspondence among the English grapheme", Phoneme" and the Korean grapheme". The machine transliteration part is com-posed of "generating pronunciation" step and "generating transliteration" step. The generating pronunciation step generates most probable correspondence between an English pronunciation unit and a phoneme. Based on the pronunciation of the English word, a Korean word is generated in "generating transliteration" step. This word and character accuracy reported for the system is 90.82% and 56% respectively.

Lee et. al. has developed an English Chinese language transliteration system [11] based on statistical approach . In the proposed model the back transliteration problem is solved by finding the most probable word E, given transliteration C. The back-transliteration probability of a word E is written as P(E|C) as stated by Bayes' rule. In the preprocessing phase a sentence alignment procedure is applied to align parallel text at the sentence level in order to find the corresponding transliteration for a given source word in a parallel corpus. Then tagging is done to identify proper nouns in the source text. In the second step, the model is applied to isolate the transliteration in the target text. The transliteration model is further augmented with linguistic processing, to remove superfluous tailing characters in the target word in the post processing phase.

Malik A. had explained a simple rule based transliteration system for Shahmukhi to Gurmukhi script [12]. For transliteration of Shahmukhi to Gurmukhi, the PMT system uses transliteration rules. It preserves both the phonetics as well as the meaning of transliterated word. PMT is a system in which each word is transliterated across two different writing systems being used for same language. Two scripts are discussed and compared. For the analysis and comparison, both scripts are subdivided into different groups on the basis of types of characters e.g. consonants, vowels, diacritical marks, etc. Transliteration rules are then developed for character mappings between Shahmukhi and Gurmukhi. The system was tested for both classical and modern literature. The classical literature comprises of hayms of Baba Nanak, Heer by Waris Shah , Hayms by Khawaja Farid and Saif-ul-Malooq by Mian Muhammad Bakhsh. The modern literature is collected from poetry and short stories of different poets and writers. The system has reported 98% accuracy on classical literature and 99% accuracy on modern literature.

Harshit Surana and Anil Kumar Singh in 2008, proposed a transliteration system on two Indian languages Hindi and Telugu [13]. In their experiment, a word was first classified as Indian or foreign using character based n - grams. The probability about word's origin was computed based on symmetric cross entropy. Based on this probability measure, transliteration was performed using different techniques for different classes (Indian or foreign). For transliteration of foreign words, the system first used a lookup dictionary or directly map from English phoneme to IL letters. For transliteration of Indian word, the system first segmented the word based on possible vowels and consonant combinations and then mapped these segments to their nearest letter combinations using some rules. The above steps generate transliteration candidates which were then filtered and ranked using fuzzy string matching in which the transliteration candidates were matched with the words in the target language corpus to generate target word. The out of vocabulary words are not handled by this system.

Hong et al. have developed a Hybrid Approach to English-Korean Name Transliteration system [14]. The base system is built on "MOSES? with enabled factored translation features. The process of transliteration begins by mapping the units of source words to units of target words. The base

system is expanded by combining various transliteration methods viz. web based n-best re ranking, a dictionary based method, and a rule-based method. The pronouncing dictionary is created from an English-Korean dictionary containing 130,000 words and CMU pronouncing dictionary containing over 125,000 words and their transcriptions. For a given English word, if the word exists in the pronouncing dictionary, then its pronunciations are translated to Korean graphemes by a mapping table. Also 150 rules have created to map English alphabet into one or more several Korean graphemes. The system achieved 45.1 and 78.5, respectively, in top-1 accuracy.

P.J. et. all. proposed English to Kannada transliteration system [15] using Support Vector Machine. The proposed system uses sequence labeling approach for transliteration which is a two step approach. The first step performs segmentation of source string into transliteration units and the second step performs comparisons of source and target transliteration units. It also resolves different combination of alignments and unit mappings. The whole process is divided into three phases: preprocessing, training using SVM and transliteration. The preprocessing phase converts the training file into a format required by SVM. The authors are using database of 40,000 Indian place names for the training of SVM. In this phase, English names are romanized and then segmented based on vowels, consonants, digraph and trigraphs. Alignment is performed at the end of the preprocessing phase. During training phase, aligned source language names are used as input and target language names are used as label sequence and given to SVM. The training phase generates a transliteration model which produces top N probable Kannada transliteration during transliteration phase. The system is tested on 1000 out of corpus place names. The system is also compared with Google Indic system and reported higher accuracy while transliterating Indian names and places. The overall accuracy of the system is 87.28%.

Kak et al. have developed A rule based converter for Kashmiri language [16] from Persio-Arabic to Devanagari script. As Devanagari letters do not have one to one correspondence with Persio-Arabic characters. So character position and the combination of the characters were also taken into consideration while developing the rules. The converter was tested on 10000 words and more than 90% accuracy was found.

Deep and Goyal have developed a Rule based Punjabi to English transliteration system for common names [17]. The proposed system works by employing a set of character sequence mapping rules between the languages involved. To improve accuracy, the rules are developed with specific constraints. This system was trained using 1013 preson's names and tested using different person names, city names, river names etc. The system has reported the overall accuracy of 93.22%.

Jasleen and Josan have proposed a statistical model for English to Punjabi machine transliteration of out-of-vocabulary words using MOSES, a statistical machine translation tool [18]. Letter to letter mapping is used as a baseline method in the proposed system. The problems of baseline method like multiple mappings of a character in target language or a character having no mapping in the target script are handled using statistical machine transliteration approach. The system was tested on 1000 entries. The baseline model produce 73.13% accuracy rate. The statistical method shows the improvements in performance by producing 87.72% accuracy rate.

Dhore et al. proposed Hindi to English transliteration of Named entities using Conditional random Fields [19]. Indian places names are taken as input in Hindi language using Devanagari script by the system and transliterated into English. The input is provided in the form of syllabification in order to apply the n-gram techniques. This syllabification retains the phonemic features of the source language Hindi into transliterated form of English. The aim is to generate transliteration of a named entity given in Hindi into English using CRF as a statistical probability tool and n-gram as a feature set. The proposed system was tested using bilingual corpus of 7251 named entities created from web resources and books. The commonly used performance evaluation parameter was "word accuracy?. The system has received very good accuracy of 85.79% for the bi-grams of source language Hindi.

Lehal and Saini presented an Urdu to Hindi transliteration system [20]. The system uses various

rules and lexical resources such as n-gram language models to handle challenges like multiple/zero character mappings, missing diacritic marks in Urdu, multiple Hindi words mapped to an Urdu word etc. The proposed system is divided into Pre-Processing, Processing and Post-processing stage. The preprocessing stage normalizes and joins the broken Urdu words in order to prepare them for transliteration. In the processing phase corresponding to an Urdu word, Number of possible Hindi words is generated using a hybrid system based on rule based character mapping table between Urdu and Hindi characters and a trigram character Language Model. The post-processing stage joins the broken words in Hindi and chooses the best alternative, where ever multiple alternatives for Hindi words exist. The system has been tested on 18403 Urdu words and accuracy reported was 97.74%.

Rathod et al. have proposed the named entity transliteration for Hindi to English and Marathi to English language pairs using Support Vector Machine (SVM) [21]. The overall architecture of proposed system is divided into three phases viz. Preprocessing, Training and testing. In the preprocessing phase the source named entity is segmented into transliteration units through the process of syllabification and segmented units are phonetically mapped to target language transliteration units using some rules. During training phase, the parallel data obtained during syllabification is arranged in required format and n-gram features are used to train this data. The classification is done by using the polynomial kernel function of Support Vector Machine (SVM). The system was tested for person names, historical place name, city names of Indian origin. The overall accuracy of the system recorded to be 86.52%.

Malik et al. have developed a system for transliterating Urdu words to Hindi based on statistical approach [22]. The proposed system solves the problem of Urdu-Hindi transliteration through Statistical Machine Translation (SMT) using a parallel lexicon. From the parallel Urdu - Hindi entries, two types of alignments viz. character and cluster alignments are produced. Based on the alignments 8 types of Urdu-Hindi transliteration models are developed. Two types of target language models have developed i.e. Word language model and Sentence language model scoring the well-formedness of different translation solutions produced by the translation model. By combining transliteration models based on the alignments and language models based on monolingual Urdu and Hindi corpus total 24 Statistical Transliteration (ST) systems are developed. The system has achieved the maximum word-level accuracy of 71.5%. The maximum word-level accuracy is 77.8% when the input Urdu text contains all necessary diacritical .At character-level; transliteration accuracy is more than 90%.

Sanjanashree and Anand Kumar presented a framework for bilingual machine transliteration for English and Tamil based on deep learning [23]. The system uses Deep belief Network (DBN) which is a generative graphical model. The transliteration process consists of three steps viz. Preprocessing, Training using DBN and testing. The preprocessing phase does the Romanization of Tamil words. The data in both languages is converted to sparse binary matrices. Character padding is done at the end of every word to maintain the length of the words constant while encoding as sparse binary matrices. Deep Belief Network is a generative graphical model made up of multiple layers of Restricted Boltzmann Machine, a kind of Random Markov Field and Boltzmann Machine. The system uses two layers RBM on source and target side called as source and target encoders. The sparse binary matrices act as input for source and target encoders which are trained separately. Two layers RBM on the right side is the encoders for source language and the left side is the target language encoders. The joint layer concatenates the outputs of the source and target encoders. It is the transliteration layer as at this layer transliteration takes place. DBN layers are trained using unsupervised learning algorithm called Contrastive Divergence (CD). The rate of learning for English and Tamil is 0.6 and 0.4. Back propagation is performed at the end to fine-tune the weights. A source language word is passed to source encoder to joint layer and goes through target encoders giving final output as transliterated word. For evaluation purpose, 3900 proper nouns including person names and place names in Tamil and equivalent transliterated word in English are used. 900 words are used for evaluation and rest 3000 words are used for training. The accuracy achieved is about 79%.

Lehal and Saini have also developed "Sangam: A Perso-Arabic to Indic Script Machine Transliteration Model" [24]. Sangam is a hybrid system which combines rules as well as word and character level language models to transliterate the words. The system has been successfully tested on Punjabi, Urdu and Sindhi languages and can be easily extended for other languages like Kashmiri and Konkani. The transliteration accuracy for the three scripts ranges from 91.68% to 97.75%, which is the best accuracy reported so far in literature for script pairs in Perso-Arabic and Indic scripts.

Mathur and Saxena have developed a system for English-Hindi named entity transliteration [25] using hybrid approach. The system first processes English words to extract phonemes using rules. After that statistical approach converts the English phoneme to equivalent Hindi phoneme. The authors have used Stanford's NER for name entity extraction and extracted 42,371 name entities. Rules were applied to these entities and phonemes were extracted. These English phonemes were transliterated to Hindi and a knowledgebase of English-Hindi phonemes was created. The probabilities are generated on the knowledgebase using ngram probability model. Once all the English phonemes have been transliterated, Hindi phonemes are combined to form a Hindi word. The system was tested on 1000 sentences containing 9234 name entities. The accuracy of the system was compared with human translator transliterating these name entities manually. The system attained accuracy of 83.40% as it can transliterate Person, Location, Date and Time but most of the entities of type organization are not transliterated accurately.

Sunitha and Jaya proposed a phoneme based model for English to Malayalam transliteration [26]. The system is based on pronunciation and uses a pronunciation dictionary. The proposed system takes a text as an input and split it into words. These English words are transformed into English phonemes. 39 general phonemes have been identified based on CMU dictionary to convert English graphemes into phonemes. Pronunciation dictionary stores the pronunciation of each English word so corresponding pronunciation of each English words is taken from this dictionary. The pronunciations obtained from dictionary are searched in a mapping table to obtain Malayalam graphemes using handcrafted rules. Malayalam graphemes are grouped to form Malayalam word. The proposed system suffers with Out of vocabulary words. For such cases, this system does grapheme based transliteration and directly transliterates the English graphemes to Malayalam graphemes.

Adeel and Iqbal have presented a dictionary based solution for transliterating English words to Urdu in which accent conversion problem has been solved through soundex algorithm [27]. The authors have integrated their work in existing Urdu transliteration system. For acquiring transcriptions of English words, CMU pronunciation dictionary has been used. Two step-coding, forward and backward coding has been done. Backward coding maps the Urdu alphabets to English characters and forward coding maps the English transcriptions to phonetically similar codes. For forward coding, English transcriptions have been divided into two groups. One group contains transcriptions having one to one mapping to Urdu script and the other group transcriptions have multiple mappings in Urdu script. By using these two types of coding, the authors have created a dictionary containing English words, its transcription and code. When user inputs a word, its code from the dictionary are fetched and mapping to Urdu script is done using rules defined by backward coding. The system has been evaluated by five persons. Each one of them was required to convert 3 different paragraphs against which system results have been validated. The accuracy of the system comes out to be 87%.

Balabantaray and sahoo have implemented a transliteration system for Odia-English and Odia-Hindi language pair using Moses engine [28]. The authers have used a phrase based SMT techniques for the task of transliteration. Syllable split based and character split based training models have been used for training and creating Moses language model. The parallel corpus for syllable split based model has been created using 50900 entries and using 1, 10,000 entries for character based split model. Both the training models have been tested for Odis to English and Odia to Hindi using the test data created manually of 500 entries. The accuracy of Syllable based split model for Odia to English and Odia to Hindi is respectively 89% and 86%. The accuracy of character based split model for Odia to English and Odia to Hindi is 71% and 85%.

V. CONCLUSION

In this paper work, we have presented a survey on challenges, different approaches and evaluation metrics used for different machine transliteration systems. We have also listed some of the existing transliteration systems. From the survey we have found that almost all existing language machine transliteration systems are based on statistical and hybrid approach. We have tried to list down the works of few different scholars and institutions but there might exist some more groups and organizations that are involved in the development of transliteration systems.

REFERENCES

[1] S. Karimi, F. Scholer, and A. Turpin, "Machine transliteration survey," *ACM Computing Surveys (CSUR)*, vol. 43, p. 17, 2011.

[2] M. Arbabi, S. M. Fischthal, V. C. Cheng, and E. Bart, "Algorithms for Arabic name transliteration," *IBM Journal of research and Development*, vol. 38, pp. 183-194, 1994.

[3] K. Kaur and P. Singh, "Review of Machine Transliteration Techniques," *International Journal of Computer Applications*, vol. 107, 2014.

[4] *https://en.wikipedia.org/wiki/Conditional_random_field*.

[5] *https://en.wikipedia.org/wiki/Support_vector_machine*.

[6] *https://en.wikipedia.org/wiki/Decision_tree*.

[7] *https://en.wikipedia.org/wiki/Machine_learning*.

[8] S. Wan and C. M. Verspoor, "Automatic English-Chinese name transliteration for development of multilingual resources," in *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, 1998, pp. 1352-1356.

[9] B.-J. Kang and K.-S. Choi, "Automatic Transliteration and Back-transliteration by Decision Tree Learning," in *LREC*, 2000.

[10] J.-H. Oh and K.-S. Choi, "An English-Korean transliteration model using pronunciation and contextual rules," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 2002, pp. 1-7.

[11] C.-J. Lee and J. S. Chang, "Acquisition of English-Chinese transliterated word pairs from parallel-aligned texts using a statistical machine transliteration model," in *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond-Volume 3*, 2003, pp. 96-103.

[12] M. G. Malik, "Punjabi machine transliteration," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006, pp. 1137-1144.

[13] H. Surana and A. K. Singh, "A More Discerning and Adaptable Multilingual Transliteration Mechanism for Indian Languages," in *IJCNLP*, 2008, pp. 64-71.

[14] G. Hong, M.-J. Kim, D.-G. Lee, and H.-C. Rim, "A hybrid approach to english-korean name transliteration," in *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, 2009, pp. 108-111.

[15] P. Antony, V. Ajith, and K. Soman, "Kernel method for english to kannada transliteration," in *Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on*, 2010, pp. 336-338.

[16] A. A. Kak, N. Mehdi, and A. A. Lawaye, "Building a Cross Script Kashmiri Converter: Issues and Solutions," *Proceedings of Oriental COCOSDA (The International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques)*, 2010.

[17] K. Deep and V. Goyal, "Development of a Punjabi to English transliteration system," *International Journal of Computer Science and Communication*, vol. 2, pp. 521-526, 2011.

[18] J. Kaur and G. S. Josan, "Statistical Approach to Transliteration from English to Punjabi," *International Journal on Computer Science and Engineering*, vol. 3, pp. 1518-1527, 2011.

[19] M. L. Dhore, S. K. Dixit, and T. D. Sonwalkar, "Hindi to english machine transliteration of named entities using conditional random fields," *International Journal of Computer Applications*, vol. 48, pp. 31-37, 2012.

[20] G. S. Lehal and T. S. Saini, "Development of a Complete Urdu-Hindi Transliteration System," in *COLING (Posters)*, 2012, pp. 643-652.

[21] P. Rathod, M. Dhore, and R. Dhore, "Hindi and Marathi to English machine transliteration using SVM," *International Journal on Natural Language Computing*, vol. 2, pp. 55-71, 2013.

[22] M. A. Malik, C. Boitet, L. Besacier, and P. Bhattcharyya, "Urdu Hindi machine transliteration using SMT," *WSSANLP-2013*, p. 43, 2013.

[23] P. Sanjanaashree, "Joint layer based deep learning framework for bilingual machine transliteration," in *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*, 2014, pp. 1737-1743.

[24] G. S. Lehal and T. S. Saini, "Sangam: A Perso-Arabic to Indic Script Machine Transliteration Model," in *Proceedings of 10th International Conference on Natural Language Processing*.

[25] S. Mathur and V. P. Saxena, "Hybrid appraoch to English-Hindi name entity transliteration," in *Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students' Conference on*, 2014, pp. 1-5.

[26] C. Sunitha and A. Jaya, "A phoneme based model for english to malayalam transliteration," in *International Confernce on Innovation Information in Computing Technologies*, 2015, pp. 1-4.

[27] M. A. Zahid, N. I. Rao, and A. M. Siddiqui, "English to Urdu transliteration: An application of Soundex algorithm," in *Information and Emerging Technologies (ICIET), 2010 International Conference on*, 2010, pp. 1-5.

[28] R. C. Balabantaray and D. Sahoo, "Odia transliteration engine using moses," in *Business and Information Management (ICBIM), 2014 2nd International Conference on*, 2014, pp. 27-29.