Bengali-to-English and English-to-Bengali Transliteration: A Machine Learning Based Approach

A thesis

submitted in partial fulfillment of the requirement for the Degree of

Master of Computer Science and Engineering

of

Jadavpur University

By

Soma Dey

Registration No.: 129000 of 2014-15

Examination Roll No.: M4CSE1614

Under the Guidance of

Prof. Kamal Sarkar

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

2016

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that the dissertation entitled "**Bengali-to-English and English-to-Bengali Transliteration: A Machine Learning Based Approach**" has been carried out by Soma Dey (University Registration No.: 129000 of 2014-15, Examination Roll No.: M4CSE1614) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the Degree of Master of Computer Science and Engineering. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other University or Institute.

……

Prof. Kamal Sarkar (Thesis Supervisor)
Department of Computer Science and Engineering
Jadavpur University, Kolkata-32

Countersigned

……

Prof. Debesh Kumar Das
Head, Department of Computer Science and Engineering,
Jadavpur University, Kolkata-32.

……

Prof. Sivaji Bandyopadhyay
Dean, Faculty of Engineering and Technology,
Jadavpur University, Kolkata-32.

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Approval*

This is to certify that the thesis entitled "**Bengali-to-English and English-to-Bengali Transliteration: A Machine Learning Based Approach**" is a bona-fide record of work carried out by Soma Dey in partial fulfillment of the requirements for the award of the degree of Master of Computer Science and Engineering in the Department of Computer Science and Engineering, Jadavpur University during the period of June 2015 to May 2016. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

……..

Signature of Examiner 1

Date:

……..

Signature of Examiner 2

Date:

*Only in case the thesis is approved

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled "**Bengali-to-English and English-to-Bengali Transliteration: A Machine Learning Based Approach**" contains literature survey and original research work by the undersigned candidate, as part of her Degree of Master of Computer Science & Engineering.

All information have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Soma Dey
Registration No: 129000 of 2014-15
Exam Roll No.: M4CSE1614
Thesis Title: Bengali-to-English and English-to-Bengali Transliteration: A Machine Learning Based Approach

…..……..

Signature with Date

Acknowledgement

I would like to start by thanking the holy trinity for helping me deploy all the right resources and for shaping me into a better human being.

I would like to express my deepest gratitude to my advisor, **Prof. Kamal Sarkar**, Professor, Department of Computer Science and Engineering, Jadavpur University for his admirable guidance, care, patience and for providing me with an excellent atmosphere for doing research. Our numerous scientific discussions and his many constructive comments have greatly improved this work.

I would like to thank **Prof. Debesh Kumar Das**, Head, Department of Computer Science and Engineering, Jadavpur University also **Prof. Sivaji Bandyopadhyay**, Dean, Faculty of Engineering and Technology, Jadavpur University for providing me with moral support at times of need.

I would like to specially mention Miss Promita Maitra, who always extends a helping hand at times of need. I am highly grateful to Mr. Soumya Prakash Rana who took the pivotal role in helping me in understanding the WEKA 3.6 Tool Kit, despite his other professional commitments.

I am also thankful to Miss Srijita Basu, Mr. Subhasish Pal, Miss Sayanti Mondal, Miss Mrinmoyi Pal who were always present to motivate me. Without all of them it would surely be a very lonely lab.

Most importantly none of this would have been possible without the love and support of my family. I extend my thanks to my parents, especially to my mother whose forbearance and whole hearted support helped this endeavor succeed.

This thesis would not have been completed without the inspiration and support of a number of wonderful individuals — my thanks and appreciation to all of them for being part of this journey and making this thesis possible.

……..

Soma Dey
Registration No: 129000 of 2014-15
Exam Roll No.: M4CSE1614
Department of Computer Science & Engineering
Jadavpur University

Table of Contents

List of Figures

List of Tables

ABSTRACT

Machine Transliteration has come out to be an emerging and a very important research area in the field of Machine Translation. Proper transliteration of name entities plays a very significant role in improving the quality of machine translation. In this paper, we have proposed named entity (person name, location name, organization name) transliteration from Bengali to English and English to Bengali transliteration using Support Vector Machine (SVM) and number of alternative proposed methods. Transliterating a word from the language of its origin to a foreign language is called Forward Transliteration, while transliterating a loan word written in a foreign language back to the language of its origin is called Backward Transliteration. In our methodologies, we are followed machine learning based approach in where, every unit in the source name is processed one by one from the left to the right. The classification of transliteration units (units) is done by using Support Vector Machine (SVM) with polynomial kernel function and k-Nearest Neighbor methods. After the system has been evaluated, we observed that SVM model gives the best result among the proposed models and the existing model "a Modified Joint-Source channel model" [1].

Chapter 1: Introduction

Transliteration means mapping a language from one writing system to another. The transliteration process maps a word written in a character set like the Bengali alphabet is transposed in another, say the English alphabets. It is used to translate named entities across language.

Automatic transliteration is helpful for many Natural Language Processing applications. These are following as,

- Machine Translation (MT)
- Cross Language Information Retrieval (CLIR)
- Information Extraction (IE)
- Multilingual Voice Chat application
- Search Engines

In those fields huge requirement to translate named entity from one language to another are found.

This paper addresses the problem of forward transliterating of named entity from Bengali to English and back-ward Transliteration from English to Bengali. Transliteration can be used in situations where we want to express words or concepts in a language with another script. But it is very difficult to transliterate named like a person names, location names, organization names. In

machine learning based approach, every unit in the source name is processed one by one from the left to the right and each unit is assigned a label which is a unit in target language. For a unit in the source names, a feature set is constructed and finally a unit is represented as a feature vector which is labeled with the target transliterated unit. A classifier is trained with the labeled vectors to learn the model of transliterating from a source name to the name in the target language unit by unit.

Today internet is a basic requirement for human being. So, internet users are increasing day by day, and it is very important to develop a tool to support Indian language for them. Most of the information's available in English which is familiar by only some percentage of the population. In West Bengal, most of population is not have a good knowledge in English language. As most of the information available on web or electronic information is in English, people who do not have the ability to learn English cannot make use of this electronic information without any person's help. In order to make it possible for everyone to use web based, automatic language translation is required that type of problem solve. For this type of purpose we need a translator which translate the named entities because we do not have any dictionaries which translate the named entities. We need a process or method which translates the named entities. Machine translation is one of the applications of Natural Language Processing which provide a method to translate the named entities.

In this transliteration, maps the transliteration unit (TU) from the source script to the transliteration unit (TU) of the goal script. The process of mapping mainly involves two steps:

• Segmentation of the source string into transliteration units.

Transliteration units in Bengali words take the pattern $C^{+}M$ where C represents a vowel or a consonant or a conjunct and M represents the vowel modifier or matra. The English transliteration units are of the form C*V* where C represents a consonant and V represents a vowel [1]. For mapping from the source language TUs into the target language TUs, we need the proper linguistic knowledge of the set of possible conjuncts and diphthongs in Bengali and their equivalents in English.

1.1 Mapping

We divided the mapping process into two categories based on existing models. These are following:

1.1.1 Grapheme/Spelling based mapping

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.

This mapping technique followed for Bengali language to target English language transliteration [1]. A direct orthographical mapping between Bengali and English languages that are of different

origins employing different alphabet sets, [বৃ।জ।মো।হ।ন<–>bri|j|mo|ha|n][1]. Grapheme based mapping is done between source Bengali language and target English language.

1.1.2 Phoneme based mapping

This model considers transliteration as a phonetic process rather than an orthographic process. In this mapping, 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.

Based on phonology, foreign name can usually be translated, or more appropriately transliterated into its target counterpart in terms of pronunciation similarities between them [2].

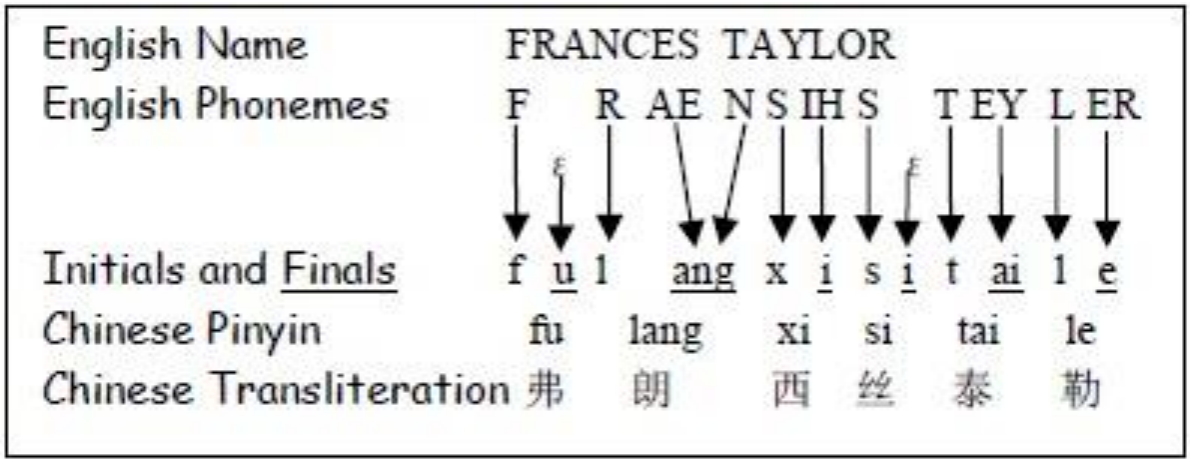

Fig. 1: English-to-Chinese transliteration Example [2]

In Fig1, English grapheme/character converted to English phoneme and then using this English phoneme generated their corresponding Chinese phoneme. And finally convert this Chinese phoneme into Chinese grapheme. Based on phonology, foreign name can usually be translated, or more appropriately transliterated into its target counterpart in terms of pronunciation similarities between them.

It is hard to extend the baseline of transliteration model by introducing additional dependencies [3], such as flexible neighboring or contextual phoneme features;

It allows only one target language phoneme to be associated with a contiguous group of source language phonemes, but not vice versa. The example in [2] exposes this limitation (see Fig. 1): /u/ and the second /i/ in the third line have to be looked as spuriously produced from dumb sound ε [4].

1.2 Major problem of transliteration

The central problem in transliteration is predicting the pronunciation of the original word. However, for languages that use different alphabet sets, the names must be transliterated or rendered in the target language alphabets. Named entities take a vital place in NLP applications. Proper identification, classification and translation of named entities are very critical in many NLP applications and pose a very big challenge to NLP researchers. Named entities are usually not found in bilingual dictionaries and they are very productive in nature. Translation of named entities is a difficult task: it involves both translation and transliteration. Transliteration is

commonly used for named entities, even when the words could be translated. Different types of named entities are translated differently.

1.3 Thesis organization

This thesis paper is organized is as follows: In **chapter 1** contain preliminary concept of machine transliteration, mapping technique of transliteration and major problem of names entity transliteration. In **chapter 2** we describe major approaches to transliteration. In **chapter 3** contain proposed methodology. **In chapter 4** contain description of data set. In chapter 5 contain evaluation and results. In **chapter 6** contain different classifiers. And at last **chapter 7** contain conclusion.

Chapter 2: Literature Survey

2.1 Major approaches to transliteration

MT (Machine transliteration) systems can be classified according to their core methodology. These are the following:

,• Rule Based Approach

• Machine learning based and Statistical based approach

2.1.1 Rule based approach

In the rule-based approach, human experts specify a set of rules to describe the translation process, so that an enormous amount of input from human experts is required. In this approach rules are created to perform the task of transliteration. Rules are created by human being

considering the properties of the source and target language. Rules-based approaches take time, money and trained personnel to make and test the rules. The main advantage of rule based approach is that if rules are properly created according to the features of both source and target language then system can transliterate those nouns also which are not present in the database. The disadvantage of rule based approach for transliteration is very difficult to implement as there are very large number of rules with various exceptions are there in this approach. These rules produce errors if they are not properly developed by human being. Another disadvantage of rule based approach is that is works only on the Indian origin names but not on the foreign names [6]. This approach used for to improve the named entity translation by combining a transliteration approach with web mining this research, using web information as a source to complement transliteration, and using transliteration information to guide and enhance web mining. A Maximum Entropy model is employed to rank translation candidates by combining pronunciation similarity and bilingual contextual co-occurrence.

Language Model, Direct Example Based, Character Sequence, Modeling, Syllable Based Model Letter To Phoneme Model (L2M) those are rule based approach.

Kang B J and Choi K S (2001) implemented the two approaches, these are 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 [7].

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 [8].

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 [9].

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 [10].

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 [11].

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 [12].

2.1.2 Machine Learning and Statistical based approach

In statistical machine translation (SMT), parallel examples are used to train a statistical translation model. Thus, it relies on statistical parameters and a set of translation and language models, among other data-driven features. This approach worked initially on a word-by-word basis. In machine learning based approach, every unit in the source name is processed one by one from the left to the right and each unit is assigned a label which is a unit in target language. For a unit in the source names, a feature set is constructed and finally a unit is represented as a feature vector which is labeled with the target transliterated unit. A classifier is trained with the labeled

vectors to learn the model of transliterating from a source name to the name in the target language unit by unit.

Some of the commonly used statistical and machine learning based approaches are 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.

This approach used by several researchers in the field of Natural Language processing. We describe are few of them.

Ekbal A et al.(2006)investigated a modified joint source channel based approach for Bengali-English. They used the regular expression to choose the transliteration units in the source word based the linguistic knowledge of possible conjuncts and diphthongs in Bengali and their equivalents in English. 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 [1].

Lee J S and Choi K S (1998) developed their systems with direct orthographical mapping from source graphemes to target graphemes. They are used the source channel model for English to Korean transliteration. They used a set of graphemes which corresponds to a source phoneme. First of all, their system segmented the English words into a set of English graphemes. After then, system produced possible set of Korean graphemes corresponding to the set of English graphemes. Finally, the system chooses the most relevant sequence of Korean graphemes. The key advantage of this technique is that, it considered a set 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 [13].

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 set. 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 bigram approach [14].

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 [15].

Goto I, Kato N, Uratani N and Ehara T (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 set 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 [16].

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 [17].

Li Haizhou, Zhang Min and Su Jian(2004) presented a method based on the joint source channel model for forward and backward transliteration. The language pair used was English-Chinese. For this English-Chinese transliteration they used noisy channel model (NCM) and Bayes rule 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 [18].

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 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 [19].

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 CRF performs better. The language pair used was English–Hindi [20].

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 [16].

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 [22].

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 [23].

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 [5].

Bhalla D 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 [24].

Chapter 3: Dataset Description

The data set contain 1000 unique named entity (person names, location name and organization name) in English from Google site and their Bengali transliteration have been stored manually. This data set divided into 10 fold cross-validation. In the open test, one fold is used for testing while the remaining 9 fold are used as the training materials. This process is repeated 10 times to yield an average result, which is called the 10-fold validation. Therefore, for simplicity, we randomly select one of the 10 fold, which consists of 900 entries, as the standard open test set to report results. In the close test, all data entries are used for training and testing. That means, according to my dada set 900 data used as a training data or training corpus and remaining 100 used as a test data. Each Bengali name contains 3 TU an average. 10 times our proposed model tested by 100 new Bengali names. Each time test data contains 300 TU an average. At last, we average the accuracy of 10 fold and get the actual accuracy of the system. For experimental purpose, we separated each Bengali name and its English transliteration into Transliteration unit. Transliteration units in Bengali words take the pattern $C^{+}M$ where C represents a vowel or a consonant or a conjunct and M represents the vowel modifier or matra. The English transliteration units are of the form C*V* where C represents a consonant and V represents a vowel [1]. So, These Bengali names are segmented into TU .But in the case of English name segmentation into TU we have been troubled some problem. This problem solves using the linguistic knowledge in the form of valid conjuncts and diphthongs in Bengali and their English representation.

Chapter 4: Proposed Methodology

Initially broad survey of various methods used for Transliteration in Indian and Foreign languages is presented. To improve the accuracy of existing transliteration system [1] we proposed a unified framework for machine transliteration, in this framework we proposed SVM model with a number of alternatives. The overall system framework of proposed methodology is depicted in Figure 2. The coding part for feature representation we used Neat Beans IDE 8.0.2 and WEKA 3.6 tool kits is used for classification of the patterns using SVM (SMO) and K-NN (IBF). This section focuses on the various steps needed to obtain the transliteration of named entity written in Bengali to English script. The overall logical flow of the transliteration system is divided into following three modules.

- Feature Extraction
- Feature representation.
- Classification

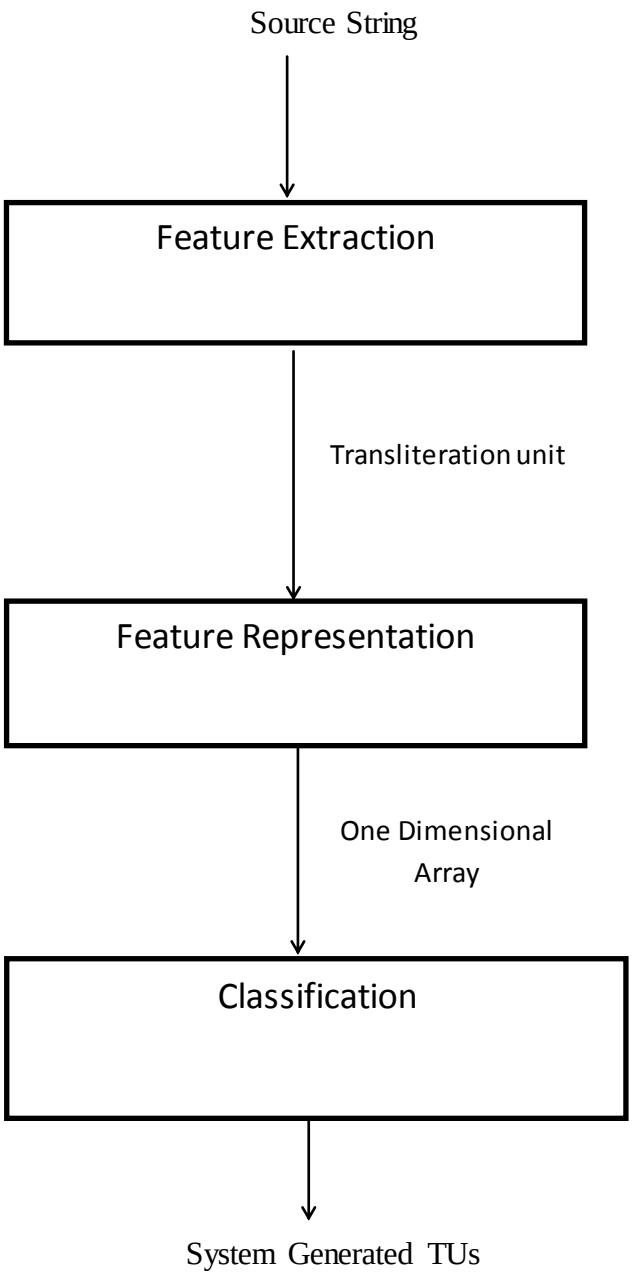

Fig. 2: System framework

4.1. Feature Extraction

In this phase, taking source string from the data set and Segmentation of the source string into transliteration unit. Transliteration units in Bengali words take the pattern $C^{+}M$ where C represents a vowel or a consonant or a conjunct and M represents the vowel modifier or matra. The English transliteration units are of the form C*V* where C represents a consonant and V represents a vowel(1). It uses the linguistic knowledge of possible conjuncts and diphthongs in Bengali and their equivalents in English for feature extraction.

Suppose that we have a Bengali name $\alpha = x_1 x_2 \ldots\ldots\ldots x_m$ and an English transliteration $\beta = y_1 y_2 \ldots\ldots\ldots y_n$ where x_i, i = 1: m are Bengali transliteration units and y_j, j = 1: n are English transliteration units. An English transliteration unit may correspond to zero, one or more than one transliteration unit in Bengali. Often the values of m and n are different.

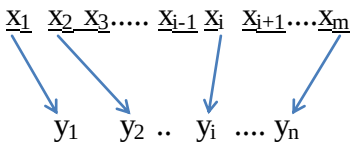

Fig. 3: Show the alignment of TUs[1]

where there exists an alignment ɣ with $\langle b,e\rangle_1 = \langle x_1,y_1\rangle$; $\langle b,e\rangle 2 = \langle x_2, y_2\rangle$; …. and $\langle b,e\rangle k = \langle x_m,y_n\rangle$. A transliteration unit correspondence <b, e> is called a transliteration pair. To generate an efficient model the TUs should be properly aligned. These features are used to train the model using support vector machine. This transliteration model then used to predict a target language word for new source language word.

Name entity transliteration is a very challenging task. In these methodologies uses the linguistic knowledge of possible conjuncts and diphthongs in Bengali and their equivalents in English for segment the words into TUs. A bilingual training corpus has been kept that contains entries mapping Bengali names to their respective English transliterations. To automatically analyze the

bilingual training corpus to acquire knowledge in order to map new Bengali names to English, We are extracted the TUs from the Bengali names and the corresponding English names, and then Bengali TUs are associated with their English counterparts.

Some examples are given below:

অনিন্দিতা(anindita)->[অ#নি#ন্দি#তা]

anindita->[a#ni#ndi#ta]

বৃথিকা(brithika)->[বৃ#থি#কা]

brithika->[bi#thi#ka]

After we retrieved the transliteration units from a Bengali-English name pair, it associates the Bengali TUs to the English TUs along with the TUs in context.

For example, it derived the following transliteration pairs or rules from the name-pair:

আশালতা(Ashalata)->Ashalata

Source language				**Target Language**	
Previous Tu	Tu	Next Tu		Previous Tu	Tu
-	আ	শা	<->	-	A
আ	শা	ল	<->	A	sha
শা	ল	তা	<->	sha	la
ল	তা	-	<->	la	ta

Transliteration is not only used for named entities, it also translated the words [লোকসভা(Lok Sava) is translated to Lok Sava(literal translation) although লোক(Lok) and সভা(Sava) are vocabulary words]. On the other hand কল্যাণী বিশ্ববিদ্যালয়(Kalyani viswavidyalaya) is translated to KalyaniUniversity in which কল্যাণী(Kalyani) is transliterated to Kalyani and বিশ্ববিদ্যালয়(viswavidyalaya) is translated to University.

4.1.1 Problems faced during mapping into TUs

- One of the problems for alignment into TUs, is the number of transliteration units retrieved from the Bengali and English words may differ. The [জগমোহন (jogmohan) <->jogmohan] name pair yields 5 TUs in Bengali side and 4 TUs in English side [জ#গ#মো#হ#ন<->jo|gmo|ha|n]. In such cases, the system cannot align the TUs automatically and linguistic knowledge is used to resolve the confusion. A knowledge base that contains a list of Bengali conjuncts and diphthongs and their possible English representations has been kept. In the above example, the TUs Jo and gmo do not have the same length. But jo is valid and gmo cannot be a valid TU in English because there is no corresponding conjunct representation in Bengali. So jmo is split up into 2 TUs j and mo, and alignment the 5 TUs as,

  [জ#গ#মো#হ#ন<->jo#g#mo#ha#n]

  Similarly, [সোমনাথ<->(somnath)] is initially split as [সো#ম#না#থ<->(so#mna#th], and since mna has the maximum length and it does not have any valid conjunct representation in Bengali. So, it splinted as

  [সো#ম#না#থ <-> so#m#na#th]

  In the following example, the number of TUs on both sides does not match [ক#ম#লি(kamli)<->ka#mli]

  In Bengali ml is a valid conjunct. In this example we observed that in the TU ml both are consonant aligned continuously and do not make a valid conjunct in Bengali respect to this example. So, TU ml separated as m and l (ml m|l).The above name pair can then be realigned as

  [ক#ম#লি(kamli)<->ka#m#li]

- In some cases, resolves the problem of alignment using the knowledge of Bengali diphthong. In the following example, [সাইমা (saima) <->sai|ma], the number of TU do not same of both sides . In this example the English TU sai have a length is greater than the other TU ma. The vowel sequence ai is a diphthong in Bengali that has two valid representations <আই,ঐ>. TU ai is splitted as a and i (ai a|i) and the first one (i.e. a) is assimilated with the previous TU (i.e. r) and finally the name pair appears as: [সাইমা (saima) <->sa#i#ma].

- The Bengali names and their English transliterations are split into TUs in such a way that, it results in a one-to-one correspondence after using the linguistic information. But in some cases there exits zero-to-one or many to- one relationship. An example of Zero-to-One relationship [Null-> h] is the name-pair[1] . In the bellow example we show that the zero to one mapping.
  [হাওড়া(Howrah)<->ho#w#ra#h].

- In some cases, the linguistic knowledge do not solves the mapping problem. From the name-pair[চরনদাস<->charandas] generated the Mapping
  [cha#ra#nda#s<->চ#র#ন#দা#স] which is not one-to-one. Then it the linguistic knowledge solve this problem by breaking up the transliteration unit as (nda->n#da) and generates the final aligned transliteration pair,

  [cha#ra#n#da#s <->চ#র#ন#দা#স]

  Since it finds out that n and da has a valid conjunct representation in Bengali but not nda. It should have been, [cha#ra#n#da#s<-> চ#র#ন#দা#স]
  Such training examples may be either manually aligned or maintained in the Direct Example base.

4.2 Feature Representation:

In this phase first we created a table which contains all distinct Bengali TUs and their corresponding English representation. So, each fold have an own table because each fold may not contain same data. In this phase make a feature for each data which is to be trained. Before make a feature for each data, we observe that after segmentation each Bengali TU has multiple representations. Like that,

a#ti#n অ#তি#ন

a#na#nda আ#ন#ন্দ

In the above examples Bengali TU "ন" have multiple representation in English such as "n" and "na".

a#si#t অ#সি#ত

a#ta#nu অ#ত#নু

As same as in the above two examples Bengali TU "ত" represented in English TU "t" and "ta".

bi#no#y বি#ন#য়

vi#ja#y বি#জ#য়

In the above two example also have multiple representation of Bengali TU "বি" as English TUs "bi" and "vi".

From this observation we conclude that each Bengali TUs have a multiple representation in English. So, need a table which contains a sample of some distinct Bengali TUs and their corresponding possible English representation. In Table 2 contain a sample of some distinct Bengali TUs and their corresponding English representation.

Table 1: A sample of some distinct Bengali TUs and their corresponding English representations

অ<->a, o
অং<->an
আ<->a, aa
ই<->e ,ei, i, ye
ঈ<->i
উ<->u

After this TUs representation, we make a feature for each Bengali TUs of the source string. For making a feature representation, first have been taken the Bengali TU sequence [র#ত#ন] divided into multiple windows. Features representation are variable sized n-grams i.e. Trigram as a window size. N-grams are generated using backward and forward movement. Table2 represent the trigram window for each Bengali TUs of রতন[র#ত#ন].

Table 2: Trigram window for each TUs of রতন [র#ত#ন]

Previous TU	TU	Next TU
null	র	ত
র	ত	ন
ত	ন	null

In the above table first row represent the window of size three for Bengali syllable র, second row represent the window for Bengali syllable ত and third row represent the window for Bengali syllable ন.

We create a matrix for each TU in the source string. Size of these matrix is 3xN. Where N is a total numbers of distinct TUs in the training data. For this matrix creation we found the exact

Bengali TUs from each fold wise table. When we found the exact Bengali TU from the table then the exact position of the TUs is labeled by one other is zero. The value of matrix is either 0 or 1. If the TUs is not find in the table then we putted zero, and we putted the value 1 in these matrixes when the TUs is find in a table. Representation of a matrix for each Bengali TUs র, ত, ন is given in Table3, Table4 and Table5.

Table 3: Contain matrix for TU র

Syllable	f1	f2	f3	f4	f5	f6	……	fi	fi+1	….	fn-1	fn
Null	0	0	0	0	0	0	……	0	0	……	0	0
র	0	1	0	0	0	0	…….	0	0	……	0	0
ত	0	0	0	0	1	0	…….	0	0	…….	0	0

Table 4: Contain matrix for TU ত

Syllable	f1	f2	f3	f4	f5	f6	……	fi	fi+1	….	fn-1	fn
র	0	1	0	0	0	0	……	0	0	……	0	0
ত	0	0	0	0	1	0	…….	0	0	……	0	0
ন	0	0	0	0	0	0	…….	1	0	…….	0	0

Table 5: Contain Matrix for TU ন

Syllable	f1	f2	f3	f4	f5	f6	……	fi	f_{i+1}	….	fn-1	fn
ত	0	0	0	0	1	0	……	0	0	……	0	0
ন	0	0	0	0	0	0	…….	1	0	……	0	0
Null	0	0	0	0	0	0	…….	0	0	…….	0	0

This 3xN dimension matrix is converted to one dimensional array for each TU. This one dimensional array is called feature vector for each TU.

Table 6: Vector/feature representation for each Bengali TUs.

	f1	f2	……	fn	f1	f2	……	fn	f1	f2	……	fn
র	0	0	…….	0	0	0	……	0	0	0	……	0
ত	0	0	…….	0	0	0	……	0	0	0	……	0
ন	0	0	…….	0	0	0	……	0	0	0	…….	0

This one dimensional array is required for classification phase.

4.3 Classification:

In this section describes classification details of Bengali to English machine transliteration and English to Bengali back-transliteration using WEKA tool. We used SVM and KNN as a machine learning algorithm for the classifications of patterns based on variable n-gram sizes. For this classification we created two file which contain feature or vector for each TU of test data and training data in WEKA required format. This feature or vector is rearranged as WEKA required format and Where each column separated by comma and last column represent a label of English TU which is corresponding of Bengali TU because Bengali Unicode not understand by WEKA. In a similar way, we have created another two files for back-transliteration. These files are run on WEKA tool kit using the classifier SVM(SMO) and K-NN(IBF).After this classification we getting the system generated TUs and percentage of TUs in the test data are matched. After this classification, we calculated the word agreement ratio by comparing system generated TUs with TUs in the source string.

Chapter 5: Classifiers

Two classifiers is used in these methodologies. One is SVM (Support Vector Machine) and another is K-Nearest Neighbor.

5.1 SVM:

SVM recently become one of the most popular statistical supervised learning mechanisms to obtain the transliteration .we chooses SVM because it classified large number of feature. SVM also allows nonlinear mapping if the data set is not linearly separable in a high dimensional space. In this case, it uses a non-linear kernel function for constructing the new feature space.
SVM can be used for multiclass data set where number classes can be k. In case of binary Classifier, one dimensional plane is divided in two subspaces while for multiple classes; it divides the hyper plane in multidimensional subspaces. In this classification used polynomial kernel. Then a detailed analysis of our approach is given which concludes that SVM suits the most for the task of transliteration. It learns from a set of inputs values with the associated output values. It constructs a hyper plane between two classes using binary classifier. Basically SVM 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 maximizes the separation between two classes using the method called as max margin. Due to max margin separation error rate gets minimized and if

any new input with unknown label arrives for classification, the chances of making error is minimized.

Let the data set is $\{x_1, x_2, ...x_n\}$ and the desired output or class label is $y_i \in \{+1,-1\}$, then two boundary planes and hyper plane is obtained by using following equations eq(1), eq(2) and eq(3).

$$W^T X_i - \gamma >= +1 \qquad (1)$$

$$W^T X_i - \gamma >= -1 \qquad (2)$$

$$W^T X_i - \gamma >= 0 \quad \text{where } 1 <= i <= n \qquad (3)$$

The data points should satisfy the equation eq(1), eq(2) and eq(3) for correct classification. The decision boundary can be calculated with the following optimization problem

$$\text{Minimize } \tfrac{1}{2} \|W\|^2$$

In few cases application allows misclassifications where small amount of error is tolerated. In such cases, the degree of misclassification can be measured by using the slack variable ξ and C as a control parameter. After introducing slack variable, equations eq(1), and eq(2) can be written as

$$W^T X_i - \gamma >= +1 - \xi \qquad (5)$$

$$W^T X_i - \gamma >= -1 + \xi \qquad (4)$$

Now the problem is minimized under the constraint as minimize $\tfrac{1}{2} \|W\|^2 + C \Sigma \xi i$

5.2 K-NN:

The K Nearest Neighbor Rule (k-NNR) is a very simple and intuitive method that classifies unlabeled examples based on their similarity with examples in the training set. This classifier works when we have an unlimited number of classes. The purpose of the K-Nearest Neighbors (KNN) algorithm is to use a database in which the data points are separated into several separate classes to predict the classification n of a new sample point. In this algorithm, choose k nearest neighbor from the test data set and. The way in which the algorithm decides which of the points from the training set are similar enough to be considered when choosing the class to predict for a new observation is to pick the k closest data points to the new observation, and to take the most common class among these. For choosing the most common k classes,

- First calculate the distance of test data from each class.
- Then choose the value of k.
- After then, choosing the classes which have minimum distance from the test data.

If the value of k is then no problem for choosing class but if the value of k is more than one, that time problem is arises. This problem solved by majority voting. Majority voting means which class have maximum weight among the same distance classes. This is why it is called the k Nearest Neighbors algorithm.

Chapter 6: Evaluation and Results

The systems have been evaluated for person names, historical place name, city names of Indian origin. Standard bilingual corpus in Unicode format for Hindi and Marathi is not available. 1000 NE data sets have been used for this evaluation. The system performance is measured by the terms of Transliteration Unit Agreement Ratio (TUAR) and Word Agreement Ratio (WAR) following the evaluation scheme in [1]. The evaluation parameter Character Agreement Ratio in [1] has been modified to Transliteration Unit Agreement Ratio as vowel modifier matra symbols in Bengali words are not independent and must always follow a consonant or a conjunct in a Transliteration Unit. WAR measures the correctness of the system generated word and TUAR measures the correctness of the system generated Transliteration unit. Let, B be the input Bengali word, E be the English transliteration given by the user in open test and E/ be the system generates the transliteration. TUAR is defined as, TUAR = (L-Err)/ L, where L is the total number of TUs in test dataset, and Err is the number of wrongly transliterated TUs generated by the system. WAR is defined as, WAR= (S-Err/) / S, where S is the total number of word and Err/ is the number of erroneous names generated by the system (when E/ does not match with E). Each of these models has been evaluated with linguistic knowledge of the set of possible conjuncts and diphthongs in Bengali and their equivalents in English. It has been observed that

the SVM Model with linguistic knowledge performs best in terms of Word Agreement Ratio and Transliteration Unit Agreement Ratio.

6.1 Results

We have conducted three experiments.

Experiment1: We implemented a probabilistic model which is an existing work [1] of the literature. The results of that experiment are included later.

Experiment2: We implemented a SVM model/classifier for Bengali to English transliteration and back-Transliteration.

Experiment3: We also proposed a K-NN model for this NE transliteration.

Table 7 includes the results of fold wise transliteration unit agreement ratio of all the proposed model and existing probabilistic model [1] for Bengali to English transliteration. We concluded from this table that the performance of SVM model is better than other model.

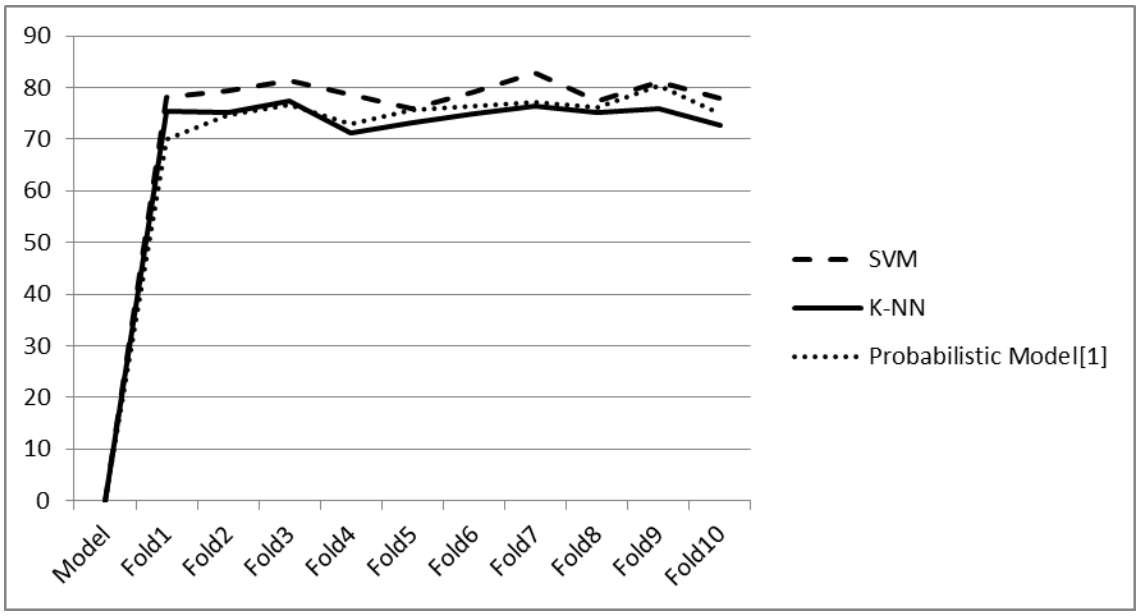

Fig. 4: Fold wise TUAR of each model (Bengali to English transliteration)

Table 7: Fold wise TUAR of each model (Bengali to English transliteration)

Model	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average
SVM	78.04	79.28	81.32	78.60	76	79.08	82.95	77.45	81	77.85	87
K-NN	75.34	75.08	77.38	71.24	73.24	74.84	76.39	75.16	76	72.63	65
Probabilistic Model[1]	69.93	74.76	76.57	72.91	75.59	76.48	77.05	76.14	80.33	74.92	68

Table 8 includes the results of fold wise word agreement ratio of all the proposed model and existing probabilistic model [1] for Bengali to English transliteration. We concluded from this table that the performance of SVM model is better than other model. Besides this, it also can be concluded that SVM model gives the best word accuracy.

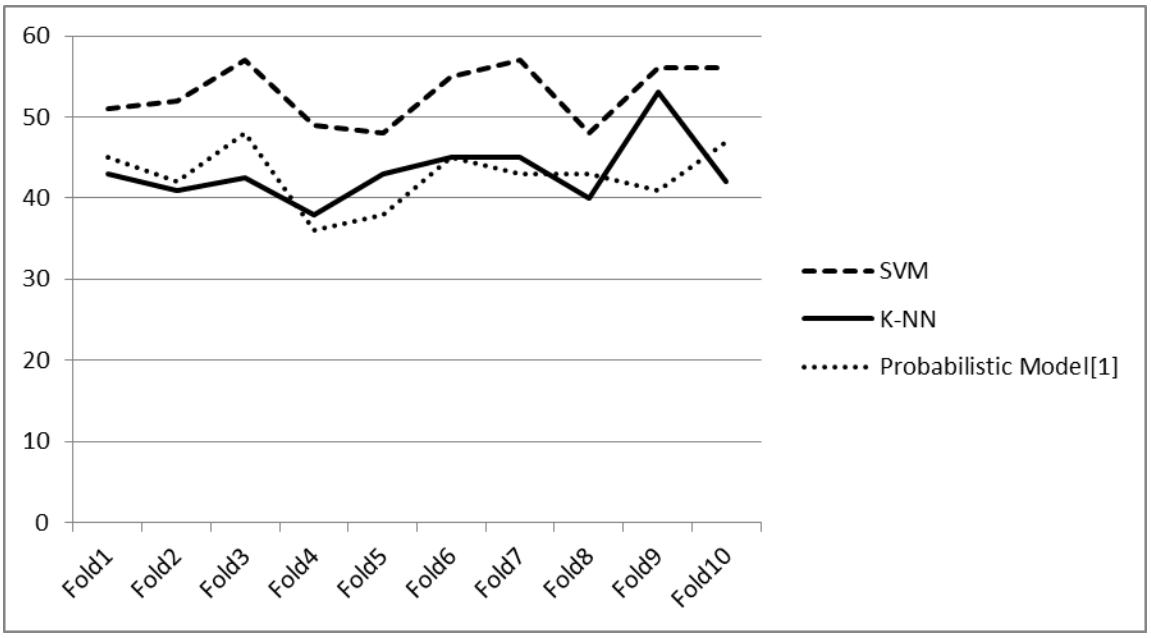

Fig. 5: Fold wise WAR of each model (Bengali to English transliteration)

Table 8: Fold wise Word Ratio for each model (Bengali to English transliteration)

Model	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average
SVM	51	52	57	49	48	55	57	48	56	56	56
K-NN	43	41	42.55	38	43	45	45	40	53	42	42
Probabilistic Model[1]	45	42	48	36	38	45	43	43	41	47	47

Table 9 includes fold wise Transliteration unit agreement ratio of all the proposed model and existing probabilistic model for English to Bengali transliteration.

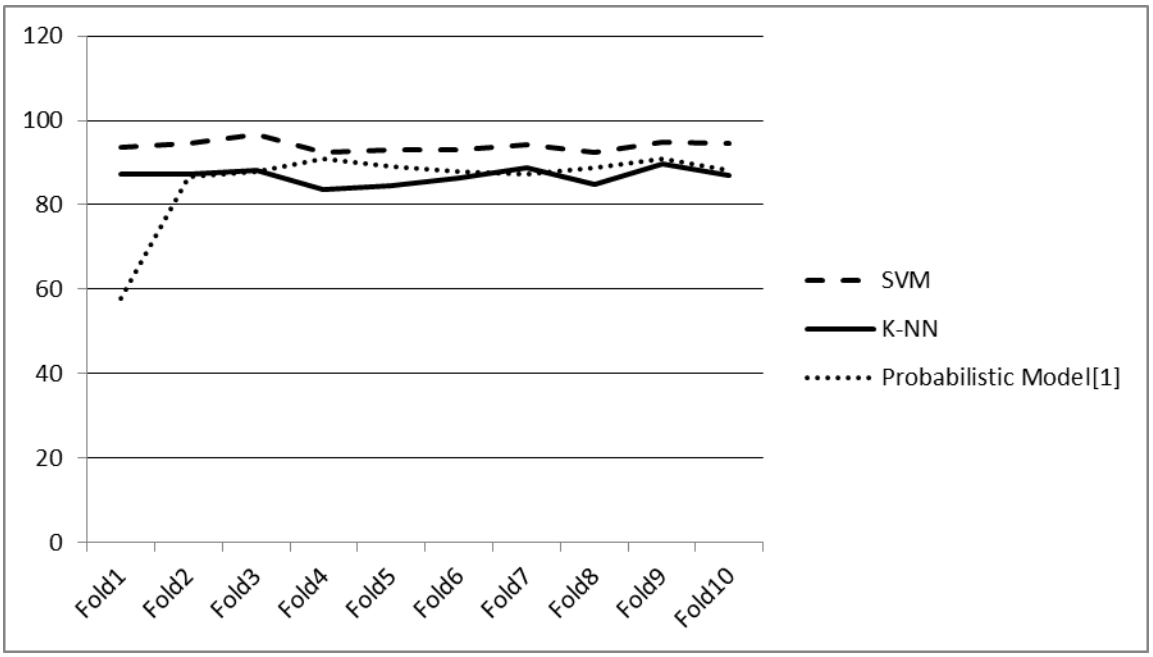

Fig. 6: Fold wise TUAR for each model (English to Bengali transliteration)

Table 9: Fold wise TUAR for each model (English to Bengal transliteration)

Model	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average
SVM	93.58	94.49	96.72	92.30	92.97	93.13	94.09	92.48	95	94.46	93.922
K-NN	87.16	87.38	88.25	83.61	84.61	86.27	88.85	84.96	89.66	86.97	86.772
Probabilistic Model[1]	57.77	86.73	87.86	90.96	88.96	87.90	87.27	88.88	91	88.27	85.56

Table 10 includes fold wise word agreement ratio of all the proposed model and existing probabilistic model for English to Bengali transliteration.

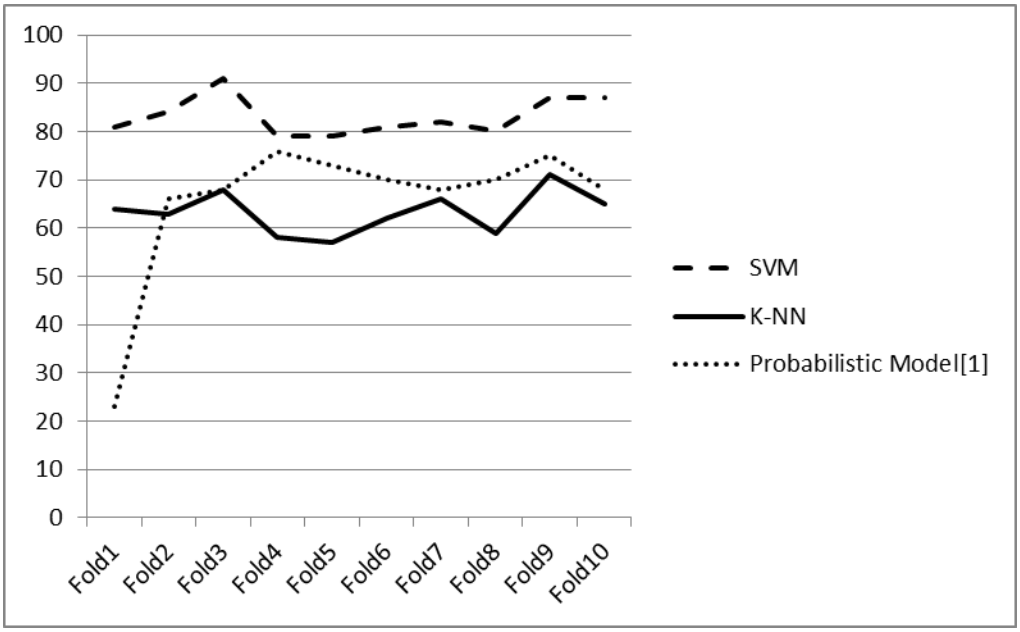

Fig. 7: Fold wise WAR of each model (English to Bengali transliteration)

Table 10: Fold wise WAR for each model (English to Bengali transliteration)

MODEL	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average
SVM	81	84	91	79	79	81	82	80	87	87	56
K-NN	64	63	68	58	57	62	66	59	71	65	42
Probabilistic Model[1]	23	66	68	76	73	70	68	70	75	68	47

Table 11 contains average TUAR of each model. We observed that from this table, SVM model TUAR better than others model.

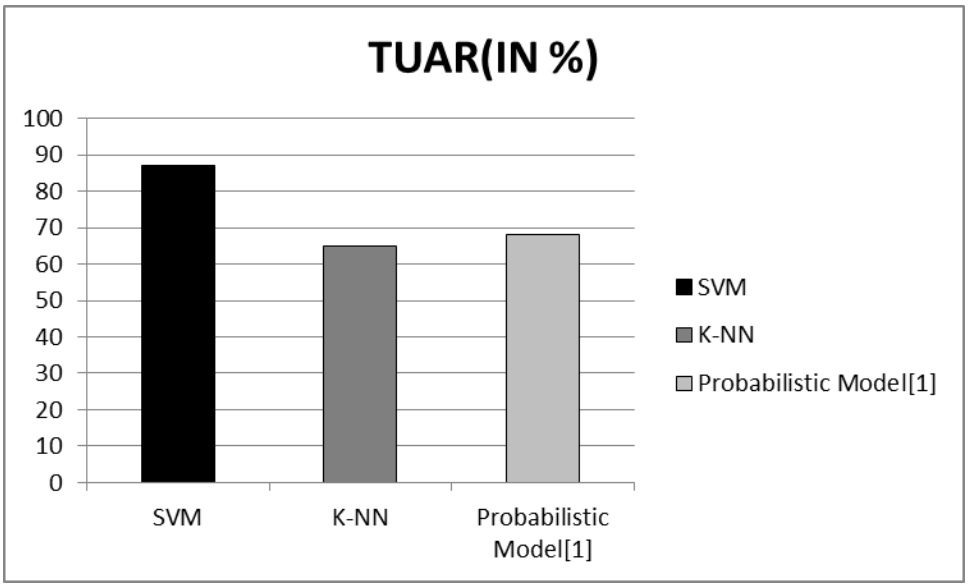

Fig. 8: Average TUAR of each model (Bengali to English transliteration)

Table 11: Average TUAR of each model (Bengali to English transliteration)

MODEL	TUAR (IN %)
SVM	87
K-NN	65
Probabilistic Model[1]	68

Table 12 contains average WAR of each model. We observed that from this table, SVM model WAR better than others model.

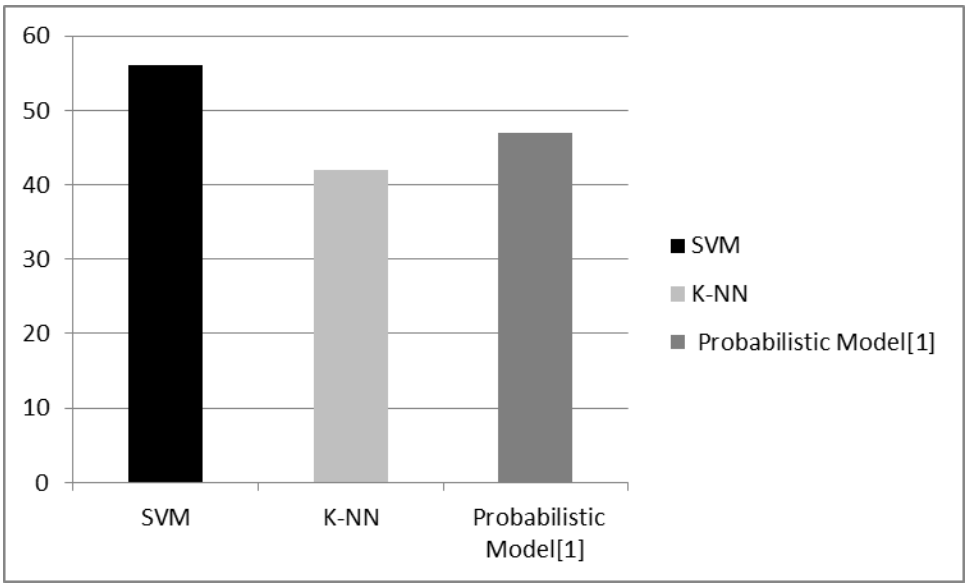

Fig. 9: Average WAR of each model (Bengali to English transliteration)

Table 12: Average WAR of each model (Bengali to English transliteration)

MODEL	WAR (IN %)
SVM	56
K-NN	42
Probabilistic Model[1]	47

Table 13 contains average TUAR of each model. We observed that from this table, SVM model TUAR better than others model.

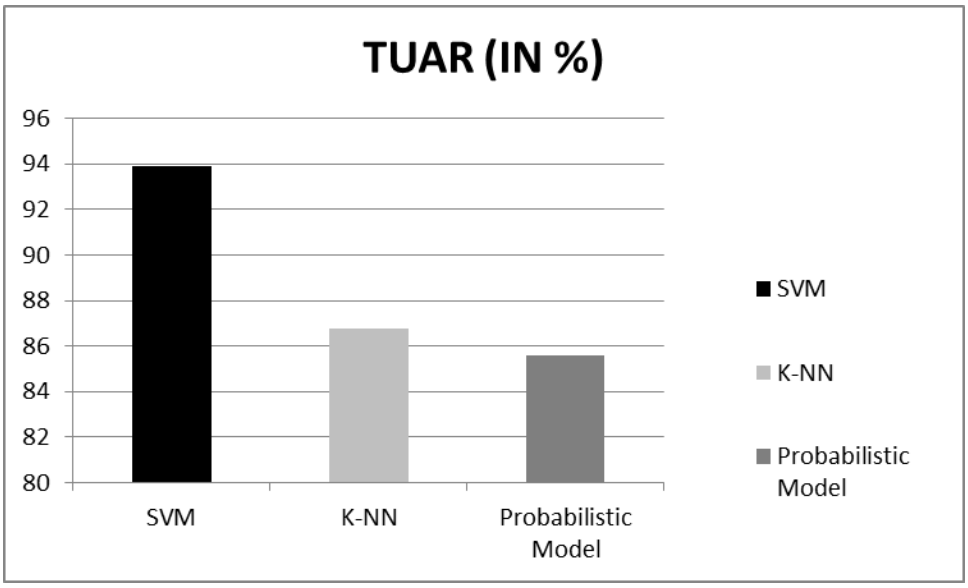

Fig. 10: Average TUAR of each model (Bengali to English transliteration)

Table 13: Average TUAR of each model (English to Bengali transliteration)

MODEL	TUAR (IN %)
SVM	93.922
K-NN	86.772
Probabilistic Model[1]	85.56

Table 14 contains average WAR of each model. We observed that from this table, SVM model WAR better than others model.

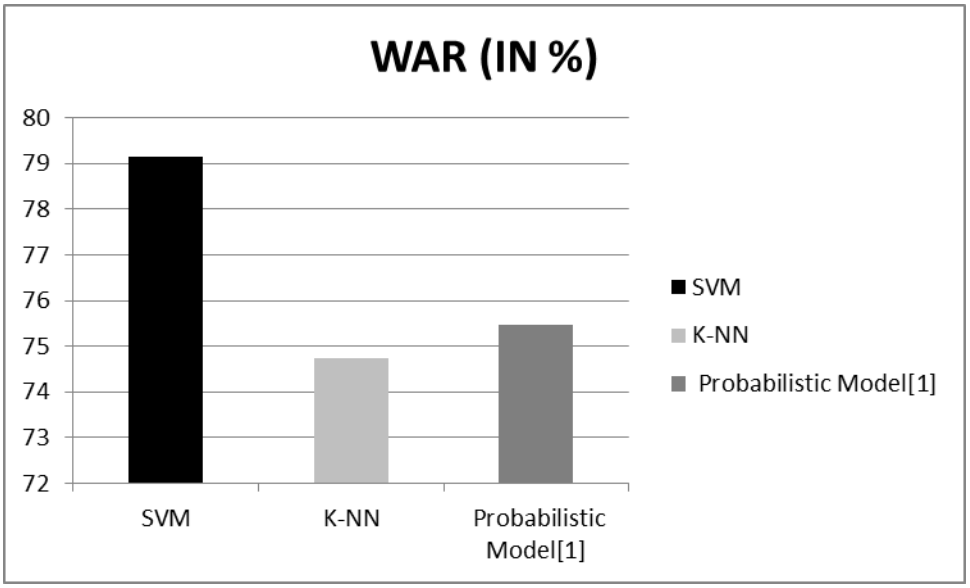

Fig. 11: Average WAR of each model (English to Bengali transliteration)

Table 14: Average WAR of each model (English to Bengali transliteration)

MODEL	WAR (IN %)
SVM	79.157
K-NN	74.73
Probabilistic Model[1]	75.468

We observed from the conducted experiments that SVM model generates the best word agreement ratio and transliteration unit agreement ratio among the proposed models, and existing probabilistic model [1]. It can be concluded from the above mentioned table 14, that SVM model word agreement ratio is 79.157% and transliteration unit agreement ratio is 93.922%, which produces better results than K-NN model and existing probabilistic model [1] for English to

Bengali transliteration. It observed from Table 11 and Table12 that SVM model word agreement ratio is 56% and transliteration unit agreement ratio is 87%, which produces better results than K-NN model and existing probabilistic model [1] than Bengali to English transliteration.

Chapter 7: Conclusion

A model of machine transliteration for Bengali to English and English to Bengali language pairs using Support Vector Machine (SVM), K-Nearest Neighbor, and probabilistic model [1] has been presented in this report. We used SVM and K-NN as a machine learning algorithm for the classification of patterns. It is desirable that transliteration model takes care of all the dependencies. SVM creates the multiple hyper planes using linear polynomial function. SVM can differentiate the adequate number of classes for all available patterns. SVM, and K-NN can produce good results when we have a lot of classes. SVM is more suitable for the transliteration task. The word agreement ratio and transliteration unit agreement ratio of SVM, K-NN, and Probabilistic model are respectively 56%, 42%, 47% and 87%, 65%, 68% for Bengali to English transliteration. Similarly, word agreement ratio and transliteration unit agreement ratio of SVM, K-NN, and Probabilistic model are respectively 79.197%, 74.73%, 75.468% and 93.922%, 86.772%, 85.56% for English to Bengali transliteration. Therefore, we concluded from our experiments, SVM is suitable for Bengali to English transliteration, and back-transliteration, and it creates less ambiguity for English to Bengali transliteration. The current system is tested for person names, place names and organization names only. It can further be extended for foreign names, organization names.

References

[1] 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.

[2] Virga P., Khudanpur S.,(2003) "Transliteration of proper names in cross-lingual information retrieval",In Proceedings of the ACL Workshop on Multi-lingual Named Entity Recognition .

[3] Och, F.J., Ney H.,(2002) "Discriminative training and maximum entropy models for statistical machine translation", In Proceedings of the 40th Annual Meeting of ACL, 295-302.

[4] Wei Gao, Kam-Fai Wong, and Wai Lam, (2004) "Phoneme-Based Transliteration

of Foreign Names for OOV Problem", International Journal on Natural Language Computing (IJNLC), LNAI 3248, pp. 110-119, 2005.

[5] Rathod H, Dhore M L, Dhore R M (2013) "Hindi and Marathi to English Machine Transliteration using SVM", International Journal on Natural Language Computing (IJNLC) Vol. 2, No.4.

[6] Kaur V, Sarao A K, Singh J "Hindi to English Transliteration System for Proper Nouns", International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6361-6366.

[7] 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.

[8] Vijayanand K., (2009) "Testing and Performance Evaluation of Machine Transliteration System for Tamil Language", Proceedings of the 2009 NEWS, pp. 48–51.

[9] 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.

[10] 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.

[11] Martin Jansche& Richard Sproat, (2009) "Named Entity Transcription with Pair n-Gram Model", Google Inc., Proceedings of the 2009 Named Entities Workshop, Singapore pp. 32–35.

[12] 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.

[13] Lee J S & Choi K S, (1998) "English to Korean Statistical Transliteration For Information Retrieval", Computer Processing of Oriental Languages.

[14] 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.

[15] Kang B J et al., (2000) "Automatic Transliteration & Back-Transliteration by Decision Tree Learning", 2nd International Conference on Language Resources and Evaluation.

[16] Goto I, Kato K, Uratani N and Ehara T, (2003) "Transliteration Considering Context Information Based on the Maximum Entropy Method", In Proceedings of MT-Summit IX, pp. 125-132.

[17] Lee J, Chang S, (2003), "Acquisition of English-Chinese Transliterated Word Pairs from Parallel-Aligned Texts using a Statistical Machine Transliteration Model", HLT-NAACL.

[18] Haizhou L, Min Z, Jian S, (2004) "A Joint Source-Channel Model for Machine Transliteration", ACL.

[19] Kumaran A et al., (2007) "A Generic Framework for Machine Transliteration", 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[20] 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.

[21] 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.

[22] Josan, G. &Kaur, J, (2011) "Punjabi To Hindi Statistical Machine Transliteration", International Journal of Information Technology and Knowledge Management, pp. 459-463.

[23] 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.

[24] 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.

[25] http://www.google.com