Statistical Approach to Transliteration from English to Punjabi

Jasleen kaur

Gurpreet Singh josan[1]

[1]Assistant Professor, Deptt of Computer Engineering
Yadvindra College of Engineering,Talwandi Sabo

Abstract--Machine transliteration plays an important role in natural language applications such as information retrieval and machine translation, especially for handling proper nouns and technical terms. Transliteration is a crucial factor in CLIR and MT. It is important for Machine Translation, especially when the languages do not use the same scripts. This paper addresses the issue of statistical machine transliteration from English to Punjabi. Statistical Approach to transliteration is used for transliteration from English to Punjabi using MOSES, a statistical machine translation tool. After applying transliteration rules average %age accuracy and BLEU score of this transliteration system comes out to be 63.31% and 0.4502 respectively.

Keywords-Machine Transliteration, Statistical approach, MOSES and Bilingual Evaluation Understudy (BLEU)

I. INTRODUCTION

Machine transliteration is an automatic method to generate characters or words in one alphabetical system for the corresponding characters in another alphabetical system [1]. Transliteration is a process that takes a character string in source language as input and generates a character string in the target language as output. The process can be seen conceptually as two levels of decoding: segmentation of the source string into transliteration units; and relating the source language transliteration units with units in the target language, by resolving different combinations of alignments and unit mappings [2]. For example, consider word in source language ‘sapna’ which is segmented into source language transliterated units ‘s’ ’a’ ’p’ ’n’ ’a’, and then these units are transliterated into target language transliterated units ‘ਸ’, ’ਪ’, ’ਨ’ and ‘ਾ’ and finally these target transliterated units into final target language word ‘ਸਪਨਾ’.

Machine transliteration is classified in two directions: forward and backward. Given a name pair (S,T) where S is the original in source language and T is the transliterated name of S in target language, forward transliteration (or transliteration) is converting from S to T; and backward transliteration(or back-transliteration) is retrieving the correct S given T. For example, if S=‘Manjeet’ is an English language string then its transliteration T in Punjabi Language is ‘ਮਨਜੀਤ’. Similarly, ‘ਮਨਜੀਤ’ in Punjabi is back transliterated into ‘Manjeet’ in English. It is in generally difficult for human to translate unfamiliar personal names, place names and names of organizations. Many technical terms and proper names, such as personal, location and organization names, are translated from one language into another language with approximate phonetic equivalents. This approximation causes the variation of words in target language and further leads to poor back transliteration.

One possible method to generate transliteration is based on the use of dictionaries, which contains words in source language and their possible transliterated forms in target language. However, this is not a practical solution since proper nouns and technical terms, which are frequently transliterated, usually have rich productivity [1]. This paper discusses another approach based on machine learning to automate the process of machine transliteration.

II. RELATED WORK

Most of the work related to English related transliteration has been done for machine translation and cross lingual information retrieval. Jong-Hoon et al. [1] presents Hybrid transliteration model which is based on both grapheme and phoneme information. Through this combination they achieved performance improvements. In this paper, they showed both grapheme as well as phoneme information is useful for machine transliteration. They showed Machine based Learning is the best machine learning method and correct pronunciation is very helpful to generate a correct Korean transliteration. Their method uses both grapheme and phoneme information in English-to-korean transliteration which achieves 13%-78% performance improvements. Nasreen and Larkey [8] had presented a method for automatically learning a transliteration model from a sample of name pairs in English and Arabic languages. In their paper, simple statistical technique for English to Arabic transliteration was evaluated. The technique learns translation probabilities between English and Arabic characters from a training sample of pairs of transliterated words from the two languages. The accuracy of this system increases with the size of the training set in both aligned conditions. Aligned training is more effective than unaligned training, and bigrams are more effective than monograms. This system was evaluated with respect to how well it can generate correct Arabic transliterations from the Arabic Proper Names dictionary for a test set of English words, after training on a non-overlapping set of word-pairs from the same source. Srinivasan et al. [15] had discussed a transliteration algorithm for mapping English named entities to their proper Tamil equivalents. This algorithm employs a grapheme-based model, in which transliteration equivalents are identified by mapping the source language names to their equivalents in a target language database, instead of generating them. The basic principle is to compress the source word into its minimal form and align it across an indexed list of target language words to arrive at the top n-equivalents based on the edit distance. The performance of this approach is also compared with a statistical generation approach using Microsoft Research India (MSRI) transliteration corpus. The results have proved that mapping is a better option than generating in the context of transliteration. The results also show that by combining compressed word format (CWF) with modified Levenshtein algorithm, increase accuracy without sacrificing precision, can be obtained. Malik A. [16] had explained a simple rule based transliteration system for Shahmukhi to Gurmukhi scripts. The Punjabi Machine Transliteration (PMT) System uses transliteration rules (character mappings and dependency rules) for transliteration of Shahmukhi words into Gurmukhi. The PMT system can transliterate every word written in Shahmukhi. It is independent of the type constraint of the word. It preserves both the phonetics as well as the meaning of transliterated word. PMT system gives more than 98% accuracy on classical literature and more than 99% accuracy on the modern literature. So PMT system fulfills the requirement of transliteration across two scripts of Punjabi. The only constraint to achieve this accuracy is that input text must contain all necessary diacritical marks for removing ambiguities. Saini and Lehal [11] present a corpus based Punjabi transliteration system from Shahmukhi script to Gurmukhi script. A corpus analysis program had been run on both Shahmukhi and Gurmukhi corpora for generating statistical data for different types like character, word and n-gram frequencies. This statistical analysis was then used in different phases of transliteration. The average transliteration accuracy of 91.37% has been obtained in this transliterated system.

III. COMPARATIVE ANALYSIS OF ENGLISH AND PUNJABI LANGUAGES

English is a West Germanic language that arose in the Anglo-Saxon kingdoms of England. It is one of six official languages of the United Nations. India is one of the countries where English is spoken as a second language. There are 21 consonant letters in English. These are B, C, D, F, G, H, J, K, L, M, N, P, Q, R, S, T, V, W, X, and Z. The rest of the letters of the alphabet are called vowels. The vowels are: A, E, I, O, U. As defined in the Constitution of India, English is one of the two official languages of communication (Hindi being the other) for India's federal government and is one of the 22 scheduled languages specified in the Eighth Schedule to the Constitution. A working knowledge of English has become a requirement in a number of fields, occupations and professions such as medicine and computing; as a consequence over a billion people speak English to at least a basic level [4].

Punjabi (ਪੰਜਾਬੀ in Gurmukhi script) is an Indo-Aryan language spoken by inhabitants of the historical Punjab region (in Pakistan and India). Punjabi is the 11th most spoken language in India and 3rd most spoken language in South Asia. The number of consonants is 41 in Gurmukhi script as given in table I. There are 19 vowels in total (10 Independent vowels and 9 dependent vowels) in Punjabi language shown in table II. Two symbols are used for nasalization (Bindi (ਂ) and Tippi (ੰ)) and one symbol (Adhak (ੱ)) that doubles the consonant before which it appears. According to the Ethnologies 2005 estimate, there are 88 million native speakers of the Punjabi language, which makes it approximately the 11th most widely spoken language in the world. In India, Punjabi is one of the 22 languages with official status in India. It is the first official language of Punjab (India) and Union Territory State Chandigarh and the 2nd official language of Haryana, Himachal Pradesh and Delhi. It is used in government, education, commerce, art, and mass media and in every day communication [5][6].

Table I Consonants in Gurmukhi script

ੳ	ਅ	ੲ	ਸ	ਹ	ਕ	ਖ	ਗ	ਘ
ਙ	ਚ	ਛ	ਜ	ਝ	ਞ	ਟ	ਠ	ਡ
ਢ	ਣ	ਤ	ਥ	ਦ	ਧ	ਨ	ਪ	ਫ
ਬ	ਭ	ਮ	ਯ	ਰ	ਲ	ਵ	ੜ	ਸ਼
ਖ਼	ਗ਼	ਜ਼	ਫ਼	ਲ਼				

Table II List of Vowels in Gurmukhi Script

Independent vowels	Dependent vowels
ਅ	ਾ
ਆ	ਿ
ਇ	ੀ
ਈ	ੁ
ਉ	ੂ
ਊ	ੇ
ਏ	ੈ
ਐ	ੋ
ਓ	ੌ
ਔ	

3.1 Problems in machine Transliteration from English to Punjabi

Transliteration is difficult in both English and Punjabi languages due to following reasons:

-Character Gap: The number of characters in, both English and Punjabi, character sets varies in both the language that makes the transliteration process difficult. The numbers of vowels are 5 and 20 [3] and numbers of consonants are 21 and 41 [3], in both English and Punjabi, respectively as explained earlier. So there is character gap in both the languages that leads to problems in transliteration process. For Example, for character **'**ਙ' in Punjabi there is no corresponding character in English.

-One-to-Multi mapping Problem: In this problem, single character in one script transform to multiple characters in another script. The Multi-mapping Problem is associated with following characters as shown in Table III. For example, character 't' in English language can be transliterated into two characters in Punjabi 'ਟ' and 'ਤ'. Some algorithm is required to select the appropriate character at different situations. Like in technical term 'bit', two transliterations are possible 'ਬਿਟ' and 'ਬਿਤ' but only first one is correct and second should be discarded by the system.

Table III Multi-mapped Characters

Characters in english	Transliterations in Punjabi
‘t’	‘ਟ’ and ‘ ਤ’
‘d’	‘ਡ’ and ‘ ਦ’
‘n’	‘ਨ’ and’ਣ’
‘r’	‘ਰ’ and’ ੜ’
‘c’	‘ਚ’ and’ਕ’

-Multi-to-One map problem: Here multiple characters in one character set leads to single character in another character set. This problem makes transliteration process difficult. This type of problem is associated with characters shown in table IV. For example, in following name, combination of two characters ‘ch’ in English language forms the single character in Punjabi language ’ਚ’.

‘Charanjeet’ → ‘ਚਰਨਜੀਤ’

Table IV Problems

Multiple characters in English	Transliterated character in Punjabi script
‘ch’	‘ਚ’
‘kh’	‘ਖ’
‘th’	‘ਥ’
‘rh’	‘ੜ’
‘bh’	‘ਭ’
‘sh’	‘ਸ਼’

-‘Double occurrence’ of certain Characters: This is similar to above explained problem. Here again two characters are clustered and mapped to single character in target language but clusters are made with ‘double occurrence’ of certain Characters in English language as shown in Table V. For example, ‘oo’ in word ‘cool’ are clustered to make one unit because they represent single character in target language. This type of clustering is associated with the following combination of characters.

Table V Double characters

'ee'
'oo'
'aa'

-Schwa sound: Schwa deletion is an important issue in grapheme-to-phoneme conversion for Indo-Aryan languages. Schwa is defined as the mid-central vowel that occurs in unstressed syllables [9]. Simple observation of Hindi words [7] provides certain information where schwa is retained and certain contexts where it is deleted without any exception. For example, name 'sapna' is transliterated into 'ਸਪਨਾ' whereby 'a' in between character 's' and 'p' is deleted and 'a' after character 'n' is retained. To find out when to retain schwa sound is not a trivial task.

-Multiple representation of same word: There can be multiple ways to write a source language word into target language. For example, for company name 'Microsoft', two representations among various transliterations in target Punjabi language are 'ਮਾਈਕਰੋਸੌਫ਼ਟ' and 'ਮਾਇਕਰੋਸੌਫ਼ਟ'. Choosing the correct one is again depends upon the perception of end user.

IV.DESIGN AND IMPLEMENTATION

The system architecture, given below in Figure I, consists of various stages through which source language text has to be passed to be converted into target language.

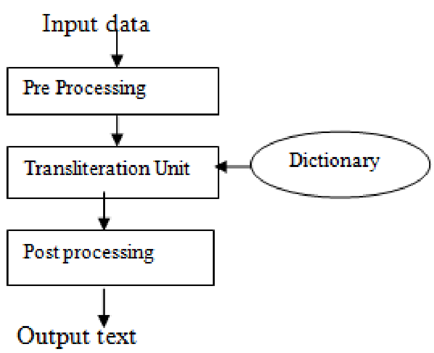

Fig I General Architecture of Transliteration system

Preprocessing: This is the first layer of the proposed model which gets an English word as input and simplifies the word by performing some pre-processing steps. At the first step, an English word goes through schwa deletion algorithm. Observations given in paper [7] are used for implementing schwa deletion algorithm with few modifications given below:

-if pair of consonants that ends in 'y', then schwa following consonants will be deleted for eg: 'kavya' will be transliterated into 'ਕਾਵ੍ਯ'.

-schwa preceding a full vowel is always retained for eg: **'udai'** will be transliterated into **'ਉਦੈ.**

-if y is preceded by syllable with vowel (e, a, o) then schwa following 'y' will be deleted. For e.g.: 'Maya 'will be transliterated into 'ਮਾਇਆ'.

-if y is preceded by syllable with vowel (i,u) then schwa following 'y' will be retained. For e.g. : 'Priya' will be transliterated into 'ਪ੍ਰਿਯ'.

The output generated from schwa deletion algorithm is passed into clustering phase. Clustering means to form groups in source language words on the basis of certain information in target language. Clusters are formed in source language because, as discussed earlier, there are certain combinations of characters in source language that are transliterated into single character in target language. For clustering, input string is divided into characters. Individual characters or tokens are extracted from both English and Punjabi strings. Tokens are separated by space character. In Text tokenization unit, string is divided into characters. Individual characters or tokens are extracted from both English and Punjabi strings. Tokens are separated by space character. Tokens generated from text tokenization unit are passed to Transliteration unit.

Transliteration module: The transliteration system is trained from the lists of proper names in English and Punjabi by using GIZA++ [12], an extension of GIZA, which determines the translation probability. Training is done with the help of 3200 names in the both English and Punjabi in tokenized form.

Post Processing: An input in target language that is Punjabi language from last unit is forwarded to the post processing unit where some post processing tasks are applied to it. Post processing tasks further improve the results using various transliteration rules. These rules are identified manually and described as follow:

Rules for a character at first position

Rule 1: If starting token of the string in English file is ‘a’, it should be transliterated into ‘ਅ’ instead of ‘ਾ’. For example name ‘ashok’ should be transliterated into **‘ਅਸ਼ੋਕ**’.

Rule 2: If starting token of string in English file is ‘i’, it should be transliterated into ‘ਇ’ instead of ‘ੀ ’ for example Iorganization name ‘infosys ‘ should be transliterated into ‘**ਇੰਫੋਸਿਸ**’.

Rule 3: If the first token in the English string is ‘y’, it will be transliterated into ‘ਯ’. For example, name of foreign origin ‘yoko’ will be transliterated into **‘ਯੋਕੋ’.**

Rule 4: If starting token of string in English file is ‘u’, it should be transliterated into ‘ਉ’ instead of ‘ੁ‘. For example name ‘umesh’ should be transliterated into ‘**ਉਮੇਸ਼**’.

Rule 5: if starting token of string in English file is ‘o’, it should be transliterated into ‘ਓ’ instead of ‘ੋ’. For example name ‘om’ should be transliterated into ‘**ਓਮ’**.

Rule 6: If starting token of string in English file is ‘e’, it should be transliterated into ‘ਏ’ instead of ‘ੇ’. For example name ‘elina’ should be transliterated into ‘**ਏਲੀਨਾ**’.

Rules for characters at last postion

Rule 7: If in the English word second last token is ‘y’ and the last token is ‘a’, then ‘ya’ will be transliterated into ‘ਯਾ’. For example name ‘maya’ will be transliterated to **‘ਮਾਯਾ’**.

Rule 8: If the second last token in English word is ‘r’ and the last token is ‘a’, ‘ra’ will be transliterated into ‘ਰਾ’. For example name ‘elora’ will be transliterated into **‘ਏਲੋਰਾ’**.

Rule 9: If the second last token in English word is ‘k’ and the last token is ‘a’, ‘ka’ will be transliterated into ‘ਕਾ’. For example, name ‘avantika’ will be transliterated into **‘ਅਵੰਤੀਕਾ’.**

Rule 10: If the last token in the string is ‘a’ and the second last token in the string is ‘i’, ‘ia’ will be transliterated into ‘ਆ’. For example, name ‘sonia’ will be transliterated into ‘**ਸੋਨੀਆ**’.

Rule 11: If the second last token in English word is ‘s’ and the last token is ’a’, then ‘sa’ will be transliterated into ‘ਸਾ’. For example, city name ‘mansa’ will be transliterated into **‘ਮਾਨਸਾ’**.

Rule 12: If the character ‘i’ is the last character in the English name and ‘a’ is the second last character in the string, means ‘ai’ sub string is present at last position in the string, then ‘ai’ will be transliterated into ‘**ਈ**’. For example city name ‘chennai’ will be transliterated into ‘**ਚੇਨਈ**’.

Rule for characters at middle of the string

Rule 13 : If token ‘a’,except at first location, is adjacent to token’ i’ in the English string,‘ai’ followed by any consonant,which is further followed by any vowel(a,u),then ‘ai’ will be transliterated into **‘ੈ’**.for example ,city name ‘jaipur’ and person name ‘naina’ will be transliterated into **‘ਜੈਪੁਰ’** and **‘ਨੈਨਾ’** respectively.

Rule14 : If two tokens ‘ai’ ,as explained in the last rule, are present at second last position,then ‘ai’ will be transliterated into ‘ੇ’. For example, English name ‘gurmail’ will be transliterated into**’ ਗੁਰਮੇਲ’**.

Rule 15: If token ‘a’,except at first location, is adjacent to token’ u’ in the English string,then ‘au’ will be transliterated into ‘ੌ’.for example word ‘laura’ will be transliterated into**’ ਲੌਰਾ’** with the usage of rule 8.

4.3 Software

We perform experiment in linux environment. The following sections describe briefly the software that was used during the project.

4.3.1 MOSES: Moses is a **statistical machine translation system** that allows to automatically train translation models for any language pair. Only translated texts (parallel corpus) is needed. An efficient search algorithm finds quickly the highest probability translation among the exponential number of choices [13].

4.3.2 GIZA++: *GIZA++* is an extension of the program GIZA which was developed by the Statistical Machine Translation team during the summer workshop in 1999 at the Center for Language and Speech Processing at Johns-Hopkins University [12]. **GIZA++** is a program for aligning words and sequences of words in sentence aligned corpus [17]. We used it to do character alignment of word-aligned pairs.

4.3.3 SRILM: SRILM is a toolkit for building and applying statistical language models (LMs), primarily for use in speech recognition, statistical tagging and segmentation. SRILM is used by Moses to build statistical language models [10].

V. EVALUATION AND RESULTS

The Evaluation of Transliteration system can be done manually or automatically by the use of metrics like Word Accuracy and BLEU. For evaluating the performance of this transliteration system, Word accuracy (WA) of the transliterated system is checked by determining the number of correct transliterations generated divided by total number of generated transliterations.

$$\text{Word accuracy} = \frac{\text{number of correct transliteration}}{\text{total number of generated transliteration}} * 100 \ \%$$

The automatic evaluation metric used for evaluating the transliteration system is BLEU Metric. **BLEU** (**Bilingual Evaluation Understudy**) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human [14].

5.1 Training and Test Data Used

In this work, we have developed and used various training data sets and testing data sets which are as follows:

Names in English Language: we collected 3844 English names. This file contains Person names, Name of Location, and Technical Terms. Out of these names, 3200 are used for training the transliteration system and 644 names are used for testing the transliteration system in case of Test Set I.

Names in Punjabi Language: we have collected 3844 Punjabi names corresponding to English names. Out of which 3200 names are used for training purpose and remaining 644 are used for evaluation of the transliteration system. The target language model is also developed from a corpus containing approximately one lakh Punjabi words from various segments like sports, business, editorials, films, health etc.

For homogeneity sake, two cases are created for training and testing this transliteration system. First test set is created by selecting 3200 names for training and remaining 644 names used for testing. In second case, 3419 names for training and remaining 425 names are used for testing. The first case is termed as SET-1 and second as SET-2.

For evaluation, the data is further categorized into various categories as follow:

-Names with Indian Origin
-Names with Foreign Origin
-Technical terms
-Names of Places (city Names)

5.2 Results

The results of statistical Transliteration system from English to Punjabi for Test set I and Test Set-II are given below in table VI:

TABLE VI Results of Transliteration System

	Set 1			Set 2			Average of Set 1 & Set 2		
	Without any preprocessing	After Schwa Deletion	After applying Rules	Without any preprocessing	After Schwa Deletion	After applying Rules	Without any preprocessing	After Schwa Deletion	After applying Rules
WA	51.86%	26.11%	65.89%	48.59%	31.23%	60.73%	50.22%	28.67%	63.31%
BLEU	0.4204	0.3307	0.4577	0.4043	0.3415	0.4428	0.4123	0.3361	0.4502

The Results on the basis of four categories are given below in table VII and Table VIII. These four categories include foreign names, Indian names, Technical Terms and City names.

TABLE VII Word accuracy rate for Category-wise results

	Set 1			**Set 2**			**Average of Set 1 & Set 2**		
Category	Without any preprocessing	After schwa deletion (%age accuracy)	After applying Rules	Without any preprocessing	After schwa deletion (%age accuracy)	After applying Rules	Without any preprocessing	After schwa deletion (%age accuracy)	After applying Rules
Indian names	68.65	31.25	80.29	63.12	38.71	68.85	65.88	34.98	74.57
Foreign Names	56.45	29.56	62.29	49.92	31.89	63.77	53.18	30.72	63.03
Technical terms	27.08	16.52	57.14	30.62	20.91	46.14	28.85	18.71	51.64
City names	55.29	27.12	63.85	50.71	33.42	64.39	53	30.27	64.12

TABLE VIII BLEU score for Category-wise results

	Set 1			**Set 2**			**Average of Set 1 & Set 2**		
Category	Without any preprocessing	After schwa deletion (%age accuracy)	After applying Rules	Without any preprocessing	After schwa deletion (%age accuracy)	After applying Rules	Without any preprocessing	After schwa deletion (%age accuracy)	After applying Rules
Indian names	0.4719	0.3701	0.5139	0.4471	0.3799	0.4759	0.4595	0.375	0.4949
Foreign Names	0.4216	0.3691	0.4432	0.3988	0.3721	0.4485	0.4102	0.3706	0.4458
Technical terms	0.3680	0.2156	0.4239	0.3691	0.2398	0.3901	0.3685	0.2277	0.407
City names	0.4201	0.3683	0.4498	0.4023	0.3743	0.4567	0.4112	0.3713	0.4532

For baseline system, Word accuracy and BLEU score comes out to be **50.22%** and **0.4123** respectively. These are results before applying schwa deletion algorithm and transliteration rules. After applying schwa deletion algorithm, Average Word accuracy and Bleu score is **28.67% and 0.3361** respectively. Word accuracy after applying Schwa Deletion Algorithm decreases because schwa Deletion algorithm is based on Hindi language which is quite different from Punjabi. Rules which are applicable on Hindi language are not followed by Punjabi language. For Example the schwa of a syllable immediately followed by a conjugate syllable is always retained [36].This rule is followed in Hindi Language but not in case of Punjabi Language. So Later on this schwa deletion Algorithm is removed from preprocessing phase. Further various transliteration rules are derived to improve the efficiency of transliteration system. Average %age accuracy and BLEU score after the application of transliteration rules is **63.31% and 0.4502** respectively.

V1. CONCLUSION

With the advent of new technology and the flood of information through the Web, it has become increasingly common to adopt foreign words into one's language. Transliteration is helpful in situations where one does not know the script of a language but knows to speak and understand the language. This paper discusses the statistical approach for transliteration from English to Punjabi using MOSES. The Efficiency of this transliteration system is evaluated manually as well as using BLEU metrics. The system is improved by applying some transliteration rules at post processing stage. It was observed that schwa deletion algorithm does not help much in improving system. This is due to the fact that this algorithm is adapted from Hindi language and all the rules which are applicable on Hindi are not applicable on Punjabi language. Average %age accuracy and Bleu score of this transliteration system without applying transliteration rules is **50.22%** and **0.4123** respectively. After applying transliteration rules, average %age accuracy and Bleu score comes out to be **63.31%** and **0.4502** respectively. Further improvements can be done in this transliteration system from English to Punjabi. One of major weakness of transliteration from English to Punjabi is dealing with multiple mapped characters as discussed earlier. Multiple-mapping leads to some problems in transliteration process. This problem also affects the accuracy of this transliteration system. Schwa sound deletion and multiple representation of same word need to be more explored.

ACKNOWLEDGMENT

I would like to express my deep and sincere gratitude to my Guide **Dr. Gurpreet Singh Josan, Assistant Professor, Yadvindra college of Engineering, Talwandi Sabo** for the continuous support of my work, for his patience, motivation, enthusiasm, and immense knowledge. I get the pleasure to present my cordial thanks for his consultation, which has been an invaluable resource throughout all levels of this work. His understanding, encouraging and personal guidance have provided a good basis for the present work.I would like to thank my family: my parents Mr.Kanwaljit Singh and Mrs.Manjeet Kaur, and my brother : Mr.Diljot Singh ,for supporting me throughout my life: and my husband, Mr.Amritpal Singh, for his encouragement and suggestions without which I would not have been in position to complete this project.Above all, I thank 'GOD' for making this mortal venture possible.

REFERENCES

[1] Jong-Hoon Oh Key-Sun Choi"Machine Learning Based English-to-Korean Transliteration using Grapheme and Phoneme Information" IEICE TRANS.INF.& SYST., VOL.E88-D, NO.7,july2005,pp 1737-1748.

[2] H.Li, M.Zhang, and J.shu "A Joint Source-channel model for Machine Transliteration", Proc.ACL2004. pp 160-167.

[3] gurmukhi script accessed from "http://www.omniglot.com/writing/gurmuki.htm" on jan 2010.

[4] English Language Accessed from "http://en.wikipedia.org/wiki/English_language", On Jan2010.

[5] Punjabi Language Accessed from "http://en.wikipedia.org/wiki/Punjabi_language" On Jan 2010.

[6] Lehal Gurpreet singh Josan Gurpreet Singh "A Punjabi to Hindi Translation System", coling 2008: Companion volume –Posters and Demonstrations, pages 157-160 Manchester, august 2008.

[7] Monojit Choudhury,Anupam Basu" A Rule Based Schwa Deletion Algorithm for Hindi" Proceedings of the International Conference On Knowledge-Based Computer Systems. 2002, pp. 343- 353.

[8] Nasreen AbdulJaleel Leah S.Larkey "English to Arabic transliteration for Cross Language Information Retrieval: A Statistical Approach"in Proceedings of the 12th international conference on information and knowledge management, 2003, pp-139-146.

[9] Monojit Choudhury,Anupam Basu and Sudeshna Sarkar" A Diachronic Approach for Schwa Deletion in Indo Aryan Languages" Association for Computations Linguistics ACL Special Interest Group on Computational Phonology (SIGPHON) Proceedings of the Workshop of the Barcelona, July 2004, pp20-26.

[10] Article SRILM accessed from http://www.speech.sri.com/projects/srilm/ on Aug 2010.

[11] Tejinder Singh Saini Gurpreet Singh Lehal , "Shahmukhi to Gurmukhi Transliteration System: A Corpus based Approach" , Advances in Natural Language Processing and Applications Research in Computing Science 33, 2008, pp. 151-162.

[12] GIZA++ accessed from "http://www.fjoch.com/GIZA++.html" on Aug 2010.

[13] Moses accessed from" http://www.statmt.org/moses/" on aug 2010.

[14] Bleu accessed from "http://en.wikipedia.org/wiki/BLEU " on dec 2010.

[15] Srinivasan C Janarthanam Sethuramalingam S Udhyakumar Nallasamy " Named Entity Transliteration for Cross-Language Information Retrieval using Compressed Word Format Mapping algorithm" In Proceedings of 2nd International ACM Workshop on Improving Non-English Web Searching (iNEWS08), CIKM-2008.

[16] M.G Abbas Malik,"Punjabi Machine Transliteration", Proceedings of the 21st International Conference on Computational Linguistics and 44th annual meeting of the ACL 2006,pages 1137-1144.

[17] Article Using GIZA++ accessed from "http://wiki.apertium.org/wiki/Using_GIZA%2B%2B" on Aug 2010.

[18] Article BLEU Accessed from http://ddi.ce.itu.edu.tr/projeler/software-center/bleu "on dec 2010.