Transliterated Search System for Indian Languages

Partha Pakray[1], Pinaki Bhaskar[2]
[1]Computer & Information Science,
Sem Sælandsvei 7-9, NO-7491, Trondheim, Norway
[2] Computer Science & Engineering,
Jadavpur University, Kolkata, India
{parthapakray, pinaki.bhaskar}@gmail.com

Abstract. The article presents the experiments carried out as part of the participation in Shared Task on Transliterated Search[1] at Forum for Information Retrieval Evaluation (FIRE) in 2013. In this shared task there were two subtasks. For Subtask 1, Trigram Model (Tri), Joint Source Channel Model (JSC), Modified Joint Source Channel Model (MJSC), Improved Modified Joint Source-Channel Model (IMJSC) have been used for transliteration. For training purpose we have used NEWS 2009 Machine Transliteration Shared Task[2] datasets. For Subtask 2, Information Retrieval (IR) purposes we have used Apache Lucene[3]. We have submitted two system results (runs) for Subtask 1 and three system results (runs) for Subtask 2. For Subtask 1, we have submitted one run for English-Bangla and one run for English-Hindi. For English-Bangla run, the system demonstrated Transliteration-Fscore of 0.1841, Eng-Fscore of 0.5768 and L-Fscore of 0.7551. For English-Hindi, the system demonstrated Transliteration-Fscore of 0.2515, Eng-Fscore of 0.7036 and L-Fscore of 0.8519. For Subtask-2, three runs of nDCG@5 is 0.2049, 0.5229, 0.5613 and nDCG@10 is 0.2073, 0.5198, and 0.5596.

1 Introduction

Transliteration is one of the important tasks in Natural Language Processing domain. The definition of transliteration is that conversion from one language script to another language script based on phonetic comparison. Lots of languages such as Indian, Russian, Arabic, etc. used Roman script to represent original language script. The central problem in transliteration is predicting the pronunciation of the original word. Transliteration between two languages that use the same set of alphabets is trivial: the word is left as it is. However, for languages those use different alphabet sets the names must be transliterated or rendered in the target language alphabets. For example, a word of Roman Script in Bengali language is as "bhalo" and its original script of Bengali language is "ভালো". Many websites (e.g. Indian Music Lyrics Websites), Social Networking (e.g. Facebook, Twitter, etc.) and blog sites have used the roman script. There are lots of challenges to convert Roman Script to Original

[1] http://research.microsoft.com/en-us/events/fire13_st_on_transliteratedsearch/default.aspx
[2] http://translit.i2r.a-star.edu.sg/news2009/corpora/
[3] http://lucene.apache.org/

Script such as spelling variation, diphthongs, doubled letters, and reoccurring constructions.

A large number of transliteration algorithm are involving Indian languages in [1, 2, 3, 4, 5]. In Asian languages transliteration algorithm are namely Chinese in [6, 7], Japanese in [8, 9], Korean in [10]; Arabic in [11, 12]; European languages in [13].

We have participated in shared task on Transliterated Search in Forum for Information Retrieval Evaluation (FIRE) 2013. In this task there were two subtasks, one is called *Subtask 1: Query Word Labeling* and another one is called *Subtask 2: Multi-script Ad hoc retrieval for Hindi Song Lyrics*. For Subtask 1, suppose that Q: W_1 W_2 W_3 …W_n is a query written in Roman script. Words, W_1 W_2 W_3 etc., could be Standard English words or transliterated from another language L (e.g. Indian Languages). The task is to label the words as E (for English Words) or L (Indian Language Script Word) depending on whether it an English word, or a transliterated L-language word. In Subtask 1 datasets were English-Hindi, English-Bangla, English-Gujarati word pairs. In Subtask 2, the input query is written in Devanagari script or its Roman transliterated form of a (possibly partial or incorrect) Hindi song title or some part of the lyrics. So the task output is a ranked list of songs in Devanagari or Roman scripts, retrieved from a corpus of Hindi film lyrics, where some of the documents are in Devanagari and some in Roman transliterated form. In Subtask 2 there was English-Hindi language pairs.

We have developed two systems (e.g. English-Bangla, English-Hindi) for Subtask 1 and one information retrieval based system for Subtask 2. The track organizers provided the training data and test data sets for both Subtask 1 and Subtask 2.

2 System for Subtask 1: Query Word Labeling

In Subtask 1 our system transliteration models [4] has used a Trigram Model (Tri), Joint Source Channel Model (JSC), Modified Joint Source Channel Model (MJSC), Improved Modified Joint Source-Channel Model (IMJSC) and International Phonetic Alphabet Based Model (IPA). An English word is divided into Transliteration Units (TUs) with patterns C*V*, where C represents a consonant and V represents a vowel. The targeted words in Indian languages are divided into TUs with pattern C+M where C represents a consonant or a vowel or a conjunct and M represents the vowel modifier or matra. The TUs are the basic lexical units for machine transliteration. The system considers the English and the Indian languages contextual information in the form of collocated TUs simultaneously to calculate the plausibility of transliteration from each English TU to various Indian languages candidate TUs and chooses the one with maximum probability. The system learns the mappings automatically from the bilingual training set being guided by linguistic features/knowledge. The output of the mapping process is a decision-list classifier with collocated TUs in the source language and their equivalent TUs in collocation in the target language along with the probability of each decision obtained from the training set. A Direct example base has been maintained that contains the bilingual training examples that do not result in the equal number of TUs in both the source and target sides during alignment. The Direct example base is checked first during machine transliteration of the input English

word. If no match is obtained, the system uses direct orthographic mapping by identifying the equivalent TU in Indian languages for each English TU in the input and then placing the target language TUs in order. The transliteration models [2] are described below where S and T denote the source and the target words respectively:

Model A: This is essentially the joint source-channel model [6] where the previous TUs with reference to the current TUs in both the source (s) and the target sides (t) are considered as the context.

$$P(S \mid T) = \prod_{i=1}^{I} P(< s,t >_i | < s,t >_{i-1}) \quad (1)$$

$$S \rightarrow T(S) = \arg\max_T \{P(T) \times P(S \mid T)\} \quad (2)$$

Model B: This is basically the trigram model where the previous and the next source TUs are considered as the context.

$$P(S \mid T) = \prod_{i=1}^{I} P(< s,t >_i | \, s_{i-1} s_{i+1}) \quad (3)$$

$$S \rightarrow T(S) = \arg\max_T \{P(T) \times P(S \mid T)\} \quad (4)$$

Model C: In this model, the previous and the next TUs in the source and the previous target TU are considered as the context. This is the improved modified joint source-channel model.

$$P(S \mid T) = \prod_{i=1}^{I} P(< s,t >_i | < s,t >_{i-1} \, s_{i+1}) \quad (5)$$

$$S \rightarrow T(S) = \arg\max_T \{P(T) \times P(S \mid T)\} \quad (6)$$

3 System for Subtask 2: Multi-script Ad hoc retrieval for Hindi Song Lyrics

In this section we will describe Information Retrieval [14] for Hindi Songs Lyrics. The Apache Lucene IR system has been used for the present task. Lucene follows the standard IR model with Document parsing, Document Indexing, TF-IDF calculation, query parsing and finally searching/document retrieval.

For this experiment we have used Hindi song lyrics corpus that was provided by FIRE Organizers[4]. So, first of all the documents had to be preprocessed. The document structure is checked and reformatted according to the system requirements. The overall system architecture for Subtask 2 is shown in Figure 1.

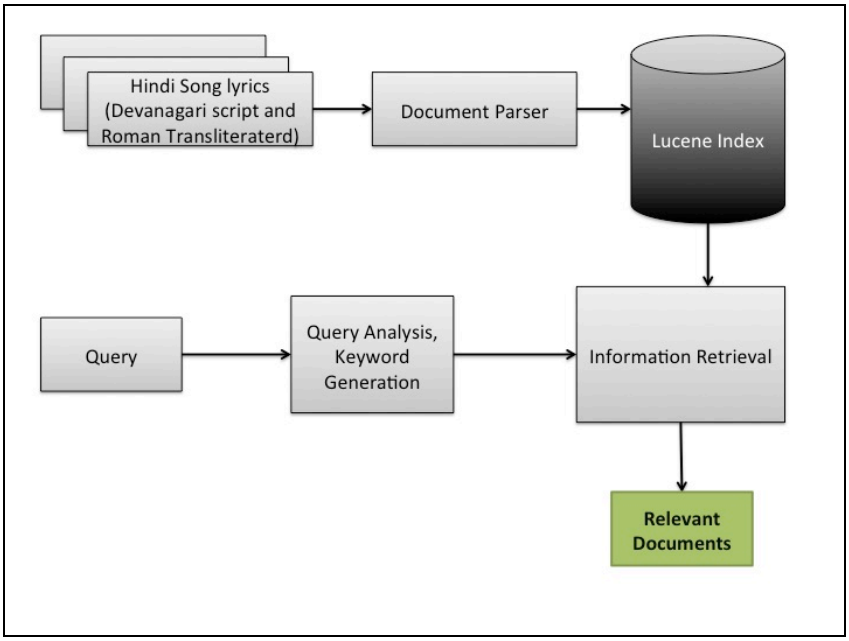

Fig. 1. System Architecture for Subtask-2

The documents were in text format. All the text test data has been parsed before indexing using text parser. The text parser extracts text from documents and removes noise. After parsing the text are indexed using Lucene.

After indexing the queries have to be processed to retrieve relevant documents. Each query is processed to identify the query words for submission to Lucene. In this step we search the documents by three ways. First way is that we searched original English roman script words of that query. Second way is that we searched Hindi transliterated words of that query that is given but output of the Subtask 1. Third way is that we just taken transliterated words and original English words of a query. This process is shown in Table 1. The query processing steps are described below:

For this experiment we have used Hindi song lyrics corpus that was provided by FIRE Organizers. All query words were searched with OR operator. OR searching retrieves at least 10 documents for each query. Now, the top 10 ranked relevant documents for each query is considered.

[4] http://research.microsoft.com/en-us/events/fire13_st_on_transliteratedsearch/default.aspx

Table 1. Query Words Generation

Way	Query	System Name
1	mann ka radio bajne de jara	NTNUNorway-1
2	मण का रेडयिो बजने दे जारा	NTNUNorway-2
3	मण का radio बजने दे जारा	NTNUNorway-3

4 Experiment on Datasets and Results

In the experiment we have used datasets from NEWS 2009 Machine Transliteration Shared Task [15] for training purpose and tested on dataset of FIRE shared task on Transliterated Search. The test result for subtask-1 has shown in Table-2 for Hindi and Table-3, Table-4 for Bangla. In Table 3 for Bangla, organizer penalize all (output) words which contain morpheme level mixing, like cinemar, torrenter, etc. as they ideally expected algorithms to split and label them as cinema\E ar\B=আর, torrent\E er\B=এর. In Table 4, organizer excludes all such words from the analysis. Hence, metric values are slightly higher than Table 3.

Table 2. Evaluation Result of Test Set for Subtask 1 (HINDI)

Language Stats (HINDI)	Metric	NTNUNorway
10 runs	Exact transliteration pairs match	540/1829
5 teams	Transliteration-precision	0.2917
#(True \H) = 2444	Transliteration-recall	0.2210
#(True \E) = 777	Transliteration-Fscore	0.2515
#(\N) = 232	Labelling accuracy	0.8025
N = Names	Eng-precision	0.5515
and ambiguities	Eng-recall	0.9717
excluded from	Eng-Fscore	0.7036
Analysis	L-precision	0.9881
	L-recall	0.7487
	L-Fscore	0.8519

Table 3. Evaluation Result (i) of Test Set for Subtask 1 (BANGLA)

Language Stats (BANGLA)	Metric	NTNUNorway
4 runs	Exact transliteration pairs match	59/242
2 teams	Transliteration-precision	0.2323
#(True \H) = 387	Transliteration-recall	0.1525
#(True \E) = 120	Transliteration-Fscore	0.1841
#(\N) = 59	Labelling accuracy	0.6897
N = Names	Eng-precision	0.4246
and ambiguities	Eng-recall	0.8992
excluded from	Eng-Fscore	0.5768
Analysis	L-precision	0.9528
	L-recall	0.6253
	L-Fscore	0.7551

Table 4. Evaluation Result (ii) of Test Set for Subtask 1 (BANGLA)

Language Stats (BANGLA)	Metric	NTNUNorway
4 runs	Exact transliteration pairs match	59/242
2 teams	Transliteration-precision	0.2389
#(True \H) = 387	Transliteration-recall	0.1525
#(True \E) = 120	Transliteration-Fscore	0.1861
#(\N) = 232	Labelling accuracy	0.6994
N = Names	Eng-precision	0.4246
and ambiguities	Eng-recall	0.9554
excluded from	Eng-Fscore	0.5879
Analysis	L-precision	0.9798
	L-recall	0.6253
	L-Fscore	0.7634

FIRE organizers provided the corpus for Hindi song lyrics; the size of this corpus is 60000 documents. In test set total 25 Hindi queries were provided by track organizers. The result of test set for subtask-2 has shown in Table-5.

Table 5. Evaluation Result of Test Set for Subtask 2

Metric	NTNUNorway-1	NTNUNorway-2	NTNUNorway-3
nDCG@5	0.2049	0.5229	0.5613
nDCG@10	0.2073	0.5198	0.5596
MAP	0.0032	0.1524	0.1970
MRR	0.0183	0.5547	0.5933

5 Conclusion

In this transliteration system we have trained our system by using NEWS 2009 Machine Transliteration Shared Task data set that was only for Named Entity datasets but in FIRE 2013 transliteration datasets which is general domain datasets that's why our system score is not good one. So, we have to train our system by general domain datasets. In future we will train our system by general domain datasets.

Acknowledgments. We acknowledge the support from Department of Computer and Information Science, Norwegian University of Science and Technology and also support from ABCDE fellowship programme 2012-1013.

References

1. Ekbal, A., Naskar, S. and Bandyopadhyay, S. 2006. A Modified Joint Source Channel Model for Transliteration. In Proceedings of the COLING-ACL 2006, 191-198, Australia.
2. Ekbal, A. Naskar, S. and Bandyopadhyay, S. 2007. Named Entity Transliteration. Interna- tional Journal of Computer Processing of Oriental Languages (IJCPOL), V olume (20:4), 289-310, World Scientific Publishing Company, Singapore.

3. Surana, H., and Singh, A. K. 2008. A More Discerning and Adaptable Multilingual Transliteration Mechanism for Indian Lan- guages. In Proceedings of the 3rd International Joint Conference on Natural Lan- guage Processing (IJCNLP-08), 64-71, India.
4. Das, A. Ekbal, Mondal, T. and Bandyopadhyay, S. English to Hindi Machine Transliteration at NEWS 2009. In Proceedings of the NEWS 2009, In Proceeding of ACL-IJCNLP 2009, August 7th, 2009, Singapore.
5. Gupta, K. and Choudhury, M. and Bali, K. 2012. Mining Hindi-English Transliteration Pairs from Online Hindi Lyrics, In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12).
6. Li, H., Min Z. and Su, J. 2004. A Joint Source-Channel Model for Machine Transliteration. In Proceedings of the 42nd Annual Meeting of the ACL, 159-166. Spain.
7. Vigra, Paola and Khudanpur, S. 2003. Transliteration of Proper Names in Cross-Lingual In- formation Retrieval. In Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-Language Named Entity Recognition, 57–60.
8. Goto, I., Kato, N., Uratani, N. and Ehara, T. 2003. Transliteration Considering Context Information based on the Maximum Entropy Method. In Proceeding of the MT-Summit IX, 125–132, New Orleans, USA.
9. Knight, K. and Graehl, J. 1998. Machine Transli- teration, Computational Linguistics, Volume (24:4), 599–612.
10. Jung, S. Y., Hong, S. L. and Paek, E. 2000. An English to Korean Transliteration Model of Extended Markov Window. In Proceedings of International Conference on Computational Linguistics (COLING 2000), 383-389.
11. Al-Onaizan, Y. and Knight, K. 2002. Named Entity Translation: Extended Abstract. In Proceedings of the Human Language Tech- nology Conference, 122– 124.
12. Al-Onaizan, Y. and Knight, K. 2002. Translat- ing Named Entities using Monolingual and Bilingual Resources. In Proceedings of the 40th Annual Meeting of the ACL, 400–408, USA.
13. Marino, J. B., R. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa and M. Ruiz. 2005. Bilingual n-gram Statistical Machine Translation. In Proceedings of the MT-Summit X, 275–282.
14. Pakray, P., Bhaskar, P., Pal, S., Das, D., Bandyopadhyay, S. and Gelbukh, A. 2010. JU_CSE_TE: System Description QA@CLEF 2010 - ResPubliQA, CLEF 2010 Workshop on Multiple Language Question Answering (MLQA 2010).
15. Li, H., Kumaran, A., Pervouchine, V. and Zhang,M. 2009. Report on NEWS 2009 Machine Transliteration Shared Task. In Pro- ceedings of ACL-IJCNLP 2009 amed Entities Workshop (NEWS 2009), Singapore.