Revisiting Automatic Transliteration Problem for Code-Mixed Romanized Indian Social Media Text

Kunal Chakma
Computer Science & Engineering Department
National Institute of Technology Agartala
Jirania, Tripura, India
kchax4377@gmail.com

Amitava Das
Human Language Technologies (HiLT) lab
University of North Texas, USA
amitava.santu@gmail.com

Abstract

Although automatic Transliteration for Indian languages is a well studied paradigm, but available transliteration techniques fail in the Indian social media context due to phenomena such as wordplay, creative spelling, code-mixing, and phonetic romanized typing; all implying that transliteration for Indian social media text has to be revisited. The paper reports an initial study on automatic transliteration for a Facebook message corpus in mixed English-Bengali-Hindi for restoration of Hindi and Bengali code-mixed words into Devanagari and Bengali script respectively.

Keywords: transliteration, code-mixing, social media text

1. Introduction

Looking at code-mixing in social media text (SMT) is overall a new research strand. SMT is characterized by having a high percentage of spelling errors and containing creative spellings (*gr8* for '*great*'), phonetic typing, word play (*goooood* for '*good*'), and abbreviations (*OMG* for '*Oh my God!*'). Non-English speakers do not always use Unicode to write social media text in their own language, frequently insert English elements (through code-mixing and Anglicism), and often mix multiple languages to express their thoughts, making automatic language detection in social media texts a very challenging task, which only recently has started to attract attention.

Different types of language mixing phenomena have, however, been discussed and defined by several linguists, with some making clear distinctions between phenomena based on certain criteria, while others use 'code-mixing' or 'code-switching' as umbrella terms to include any type of language mixing - see, e.g., Muysken (2000) or Gafaranga and Torras (2002) - as it is not always clear where borrowings/Anglicisms stop and code-mixing begins (Alex, 2008).

An essential prerequisite for any kind of automatic text processing is to be able to identify the language in which a specific segment is written. Here we will in particular address the problem of word level language identification in social media texts. Available language detectors fail for these texts due to the style of writing and the brevity of the texts, despite a common belief that language identification is an almost solved problem (McNamee, 2005). But language detection at word level is a separate problem altogether. Here in this paper we are only concentrating on transliteration.

Automatic transliteration for the Code-Mixed romanized Indian SMT is particularly problematic because there is no standard of romanization. People are quite creative in their spellings. There are various alternative phonetic spellings available for a single word. For example:

আঁখোঁ (eyes) aankhon/aankho/ankho/ankhon

ये (this) iye/yeh/ye/y

অনেক (multiple) anek/onek/onk/oneeek

অপেক্ষা (waiting) opekkha/opekha/oppekha

Even the reverse is also true. There are several cases when one Romanized word could be transliterated into multiple possible outputs based on context:

kam কাম)work /(কম) less(

aste আসতে (to come) / আস্তে (slowly)

beche বেছে (chosen) / বেঁচে (alive)

Moreover very often people mix up numerals into their Romanized phonetic representations. Those cases are even more challenging.

অচ্ছা)okay) a66a

অগোছাল (mess) ogo6alo

একটু (some) ek2

Whether transliteration for romanized word-play cases would be considered, as a restoration is an open question. For example:

bhaiiiii	ভাই
sotyiii	সত্যিই

Here "sotyiii" could be transliterated as "সত্যিই" whereas the right form is "সত্যিইই" and same for "bhaiiiii".

Transliteration of these noisy romanized words is a prerequisite in order to apply any Natural Language Processing (NLP) technique for this text genre. In this case transliteration could be described as a normalization or restoration process. Transliterated texts could be handled by existing linguistics tools like morphological analyzer, part-of-speech engine and even transliteration would be necessary if in case someone wish to explore machine translation techniques for Code-Mixed Romanized Indian SMT.

The rest of the paper is laid as follows. In the next sections, we discuss about previous works on transliteration. In the section 3 Corpus Acquisition process has been described. Section 4 details proposed transliterations models and performances are reported in the section 5. We draw our conclusions in Section 6.

2. Related Work

Transliteration is a method of transcribing one language/script word into another in a way so that the source phonetics remains preserved. Automatic machine transliteration is a well-studied paradigm. Techniques wise machine transliteration could be categorized in these three types grapheme based, phoneme based and hybrid models. A vivid details of such methods applied to various Indian languages could be found in the (Karimi et. al., 2011). All these previous works mainly talked about forward Named Entity (NE) transliterations, whereas our targeted area is open domain generic backward transliteration for noisy romanized / phonetically-typed Indian SMT. With best of our knowledge there is no similar work in this domain. Forward transliteration is a process of generating similar phonetics of a word of language A (say Hindi/Bengali) into another script of language B (say English), whereas the backward transliteration is the reverse process of getting back the word in the native script, given its transliteration in a foreign script (Gupta et al., 2012). Since there are no standard ways of spelling a word in a non-native script, transliteration content almost always features extensive spelling variations; typically a native term can be transliterated into Roman script in very many ways (Gupta et al., 2014). This rule generalization becomes more complex when the target domain is noisy SMT, as in our case.

Technique wise we are highly inspired from two previous works, as both the systems achieved high accuracies on our targeted languages Bengali and Hindi. Das et al., 2009 is based on NEWS 2009 Machine Transliteration Shared Task training datasets (Li et. al., 2009). The proposed transliteration system used the modified joint source channel (JSC) model along with two other alternatives for English to Hindi and Bengali automatic transliteration. The system also used some post processing rules for the purpose of removing the errors in the system to improve the accuracy. They performed one standard run and two nonstandard runs. Reported results showed that the performance of the standard run was better than the non-standard one. The second work is basically the participation report of the same research group in the next shared task i.e. NEWS 2010 transliteration shared task (Kumaran et. al., 2010). They proposed a transliteration technique based on orthographic rules and phoneme based approach. Phoneme based approach was based on International Phonetic Alphabet (IPA). They have submitted one standard run and two non-standard runs: while one standard and one non-standard run were submitted for Kannada and Tamil. The reported results were as follow: For the standard run, the system demonstrated means F-Score values of 0.818 for Bengali, 0.714 for Hindi, 0.663 for Kannada and 0.563 for Tamil. The reported mean F-Score values of non-standard runs are 0.845 and 0.875 for Bengali non-standard run-1 and 2, 0.752 and 0.739 for Hindi non-standard run-1 and 2, 0.662 for Kannada non-standard run-1 and 0.760 for Tamil non- standard run-1. Non-Standard Run-2 for Bengali has achieved the highest score among all the submitted runs. Hindi Non-Standard Run-1 and Run-2 runs are ranked as the 5th and 6th among all submitted Runs.

Here we tested JSC model and the IPA based model on the two different datasets. We also tested performance using trigram model, as the baseline. This is an initial experiment.

3. Corpus Acquisition

Most research on social media texts has so far concentrated on English, whereas the majority of these texts now are in non-English languages (Schroeder, 2010). Fischer (2011) provides an interesting insight on Twitter language usages in different geographical regions. Europe and South-East Asia are the most language-diverse areas of the ones currently exhibiting high Twitter usage. It is likely that code-mixing is frequent in those regions, where languages change over short geospatial distances and people generally have basic knowledge of the neighboring languages.

Here we will concentrate on India, a nation with close to 500 spoken languages (or over 1600, depending on what is counted as a language) and with some 30 languages having more than 1 million speakers. India has no national language, but 22 languages carry official status in at least parts of the country, while English and Hindi are used for nation-wide communication. Language diversity and dialect changes instigate frequent code-mixing in India. Hence, Indians are multi-lingual by adaptation and necessity, and frequently change and mix languages in social media contexts. Most frequently, this entails mixing between English and Indian languages, while mixing Indian languages is not as common. Code-mixing is much more prominent in social media than in more formal texts, as shown in the example in Figure 1, where the Bengali and Hindi segments (italics) are written in phonetic typing and not in Unicode. The last sentence is in English. This is a case of trilingual mixing.

ami eto dumb j sentence ta bujhte amr pakka 4 ghanta samay laglo.
I am too dumb that I took 4 hours to understand the sentence.
আমি এত dumb যে sentence টা বুঝতে আমার পাক্কা ৪ ঘন্টা সময় লাগলো.
tab jakey dimaag ki batti jail ...
Only then turn the light on brain …
तब जाके दिमाग कि बत्ती जली ...
I am mesmerized by your awesome sense of analogy.

Figure 1: An Example English-Bengali-Hindi Code-Mixed Message

Although we started our data collection endeavor with a motivation to collect English-Bengali code-mixed romanized SMT but after the acquisition and during the annotation process we have noticed that there is 3-4 % Hindi mixing in the data. Hindi is the national language in India and widely spoken in most of the northern parts of India and it has a strong dominance over all north India.Bengali is the national language in Bangladesh and 7th worldwide in terms of first-language speakers, whereas Hindi-Urdu is the 4th highest worldwide. So finally our collected data is code-mixed romanized SMT for the English-Bengali-Hindi languages. Two campus Facebook groups: JU Confession[1] and JU Matrimonial[2] were chosen for Bengali and Delhi University Confession [3] chosen for Bengali for the data acquisition. Total 2000 Facebook messages have been collected. Corpus statistics are reported in the Table 1. Moreover we have tested the dataset released at FIRE2014 Shared Task on Transliterated Search 2014. Although at this shared task they have released data for various other Indian languages we have used only the English-Bengali mixed data for the present experiment.

Corpus	FB		FIRE	
	HN	BN	HN	BN
Utterances	1408	1339	700	700
Words	52K	38K	24K	20K
Unique Tokens	14K	10K	7K	5K

Table 1: English-Bengali-Hindi Corpus Statistics

In table 2 we are reporting word level language. This is a trilingual code-mixed data. Here in the distribution for all the 3 languages and the distribution of universal tokens. “haha (smile)”, emoticons, punctuations, symbols and etc. are considered as a language independent/universal. FIRE2014 data is bilingual.

		Lang1 (EN)	Lang2 (BN)	Lang3 (HN)	Univ.
FB	EN-HN	42.65%	0%	36.72%	17.10%
	EN-BN	45.22%	20.75%	4.10%	1.12%
FIRE	EN-HN	44.11%	0%	38.60%	14.06%
	EN-BN	40.26%	34%	0%	17.22

Table 2: Word Level Language Distributions of English-Bengali-Hindi Corpus

[1] https://www.facebook.com/pages/JU-Confessions/210357182480508
[2] https://www.facebook.com/jumatrimonial
[3] https://www.facebook.com/duconfessions4everyone

As mentioned word level language detection is a separate challenging research problem of this text genre we invested out time to annotate word level languages. For word level language marking we finalized these 17 categories as mentioned in the Table 3. Word level mixing cases have also been noticed in our corpus so we defined all these word level categories: ne+*, acro+* and the others en+*, bn+ and hn+ categories. For better understanding we also included word level distributions (%) of each category in our corpus. In a separate recent study by us (to appear, reference removed for anonymity), we discussed mainly about the automatic word-level language detection techniques from code-mixed romanized SMT. Here in this paper we have considered that the word level languages are already given and the system has to automatically transliterate words based on language markings.

Tag (%)	Description
en (41.45)	English word
bn (35.02)	Bengali word
hi (2.70)	Hindi word
ne (1.92)	Named Entity (NE)
ne+en_suffix (0.02)	NE + Eng. suffix
ne+bn_suffix (0.08)	NE + Bng. suffix
ne+hi_suffix (0.003)	NE + Hnd. suffix
en+bn_suffix (0.08)	Eng. word + Bng. suffix
en+hi_suffix (0)	Eng. word + Hnd. suffix
bn+en_suffix (0.003)	Bng. word + Eng. suffix
hi+en_suffix (0)	Hnd. word + Eng. suffix
acro (0.20)	Acronym
acro+en_suffix (0)	Acronym + Eng. suffix
acro+bn_suffix (0.003)	Acronym + Bng. suffix
acro+hi_suffix (0)	Acronym + Hnd. suffix
univ (18.39)	Universal
undef (0.153)	Undefined / Others

Table 3: Word Level Language Tags and Their Distributions

4. Transliteration Models

Joint source channel (JSC) model is one of the most successful transliteration models for Asian-Indian languages. JSC model is originally proposed by (Hazhiou et al., 2004) and then successfully used by various others researchers (Ekbal et al., 2006; Ekbal et al., 2007; Surana and Singh, 2008). A JSC model breaks down source and target words into the smallest phoneme units called Transliteration Units (TUs) and then predict the best (*maximum probability*: $S \rightarrow T(S) = \arg\_\max_T \{P(T) \times P(S \mid T)\}$) TU for a given source by looking at contextual source and target TUs. The system learns these mappings automatically from the bilingual training set. The current system produce more than one possible ranked output for a given input word and finally the performance of the system has been measured using Mean Average Precision (MAP).

To break down TUs previous research suggested (Ekbal et al., 2006; Ekbal et al., 2007; Surana and Singh, 2008) regular expression (*C***V*), where *C* represents a consonant and *V* represents a vowel. Here the sources are phonetically typed Romanized SMT and then it is quite reasonable to use same setup (*C***V*). For the target TU breaking the rule is (*C*+*M*?), where *C* represents a consonant or a vowel or a conjunct and M represents the vowel modifier or *matra*. The system considers the linguistic knowledge in the form of conjuncts and/or diphthongs in Hindi and Bengali. It is expected that the numbers of TUs in a source-target pair would be same; otherwise that pair would be considered as an exception (see the section 4.5).

In this experimental setup three transliteration models have been tested. We started with the Trigram model as the baseline and then experimented with the Joint Source Channel model and the Modified Joint Source Channel model and finally the IPA based model. More formal definitions of these models are described as follows.

4.1 Trigram Model (TRI)

The trigram model considers ±1 source TUs as the context. The model could be defined using the following equation.

$$P(S \mid T) = \tilde{\bigcirc}_{k=1}^{K} P(<s,t>_k \mid s_{k-1}, s_{k+1})$$

Where S_{k-1} is the previous source TU and S_{k+1} is the next source TU around the to-be-transliterated source TU. K is the total numbers of TUs present in a word. $P(S \mid T)$ is the transliteration probability into the target language, for a given source.

4.2 Joint Source Channel Model (JSC)

Joint Source Channel model, proposed by Hazhiou et al., 2004 where the previous TUs in both the source and the target sides are considered as the context.

$$P(S|T) = \prod_{k=1}^{K} P(<s,t>_k | <s,t>_{k-1})$$

4.3 Modified Joint Source Channel Model (MJSC)

Modified Joint Source Channel model (MJSC) is a slight modification over JSC. The MJSC considers one more additional context i.e. S_{k+1} : the next TU of the source word over JSC.

$$P(S|T) = \prod_{k=1}^{K} P(<s,t>_k | <s,t>_{k-1}, s_{k+1})$$

4.4 International Phonetic Alphabet (IPA) Model

The International Phonetic Alphabet (IPA) is a system of representing phonetic notations based primarily on the Latin alphabet and devised by the International Phonetic Association as a standardized representation of the sounds of spoken language. The machine-readable Carnegie Mellon Pronouncing Dictionary[4] has been used as an external resource to capture source language IPA structure. The dictionary contains over 125,000 words and their transcriptions with mappings from words to their pronunciations in the given phoneme set. The current phoneme set contains 39 distinct phonemes. As there is no such parallel IPA dictionary available for Indian languages. Romanized Indian languages have been mapped to TUs in Indian languages during training.

4.5 Exceptions and Discussion

There are several cases when number of TUs in the source-target pair does not match. For example:

e_kdm एकदम

l_g ta लगता

h_ha_i है

cha_llo_w छल्लो

a_mr আ_মা_র

kmn কে_ম_ন

r আ_র

These cases are mainly noisy SMT abbreviations. These cases are directly added to the exception list. Numbers of entries in the Hindi and Bengali exception list are 10% and 8% respectively. These percentages have been calculated based on total number of entries in the total corpus.

[4] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Majority of the previous works on transliteration talked about NEs. Transliterations of NEs do not change with context, but for a general-purpose transliteration there are several cases when possible transliteration change with context. Basically homonyms. For example:

taara তারা/তাড়া

bhujte বুঝতে/বুজতে

To resolve these cases contextualization may help, rather POS may help. These cases are relatively fewer. Developing a POS tagger for this text genre is a different research problem altogether so we kept these issue unattended.

5. Performance

All results presented here are 5-fold cross validations, but before presenting results let us discuss about evaluation matrices. FIRE 2014 shared task (Roy et. al., 2013) defined Exact transliteration pair match (ETPM) for transliteration evaluation.

EPTM=#(Pairs for which transliteration match exactly)/#(Pairs for which both o/p and reference labels are L)

It is quite legitimate evaluation metric for the FIRE 2014 shared task because the task itself had two goals: word-level language detection and automatic transliteration of romanized Indian languages words. Although we have worked on the same dataset but we are only concentrating on the transliteration with a presumption that the word-level languages are known. Therefore our evaluation metrics are standard precision, recall and f-measure. Even in the NEWS shared task (Zhang et. al., 2012) Mean Average Precision (MAP) have been used to judge system performances on automatic named entity transliterations. Since a name may have multiple correct transliterations, all these alternatives are treated equally in the evaluation, that is, any of these alternatives is considered as a correct transliteration, and all candidates matching any of the reference transliterations are accepted as correct ones. Although our systems (all the models) produce multiple outputs but there is only one reference transliteration per word in the golden set, therefore MAP is not much relevant for our task. To extend our rationale let us quote (Roy et. al.,

2013). Knight and Graehl, 1998 point out, back-transliteration is less forgiving than forward transliteration for there may be many ways to transliterate a word in another script (forward transliteration) but there is only one way in which a transliterated word can be rendered back in its native form (back-transliteration). Our task thus requires the algorithm to only perform back-transliteration and thus there is only one correct transliteration answer for a word in a given context.

Here in the Table 4 we have reported performances of all the transliteration models on both the data set. Since our dataset is small we have used additional resources (Gupta et. al., 2012) to train our system. These additional resources consist 30K Hindi word pairs and 25K Bengali word pairs. Results using additional resources reported separately. As in our case each word attempted by the system for the transliteration thus precision, recall and f-measures values are same. Henceforth accuracy figures reported here in the Table 4 are f-measures.

Models	Data Set	Accuracy			
		Train		Add. Resources	
		BN	HN	BN	HN
TRIGRAM	FB	.52	.60	.55	.65
	FIRE	.51	.62	.54	.67
+JSC	FB	.57	.69	.62	.78
	FIRE	.59	.70	.66	.75
+MJSC	FB	.59	.70	.62	.78
	FIRE	.60	.71	.66	.75
+IPA	FB	.66	.79	.69	.84
	FIRE	.69	.88	.71	.90

Table 4: Transliteration Accuracies

5.1 Comparison

To compare our results let us have a look over results of participated teams at FIRE 2014 shared task. For the Bengali transliteration task transliterate-kgp achieved highest accuracy of EPTM .8 whereas the team did bad in the language detection module and therefore transliteration on detected words is not directly comparable with others teams such as IITP-TS and JU-NLP-LAB. Results of both the teams are directly comparable i.e. .67 and .62 respectively. For Hindi transliteration two teams: BITS-Lipyantaran and the IITP-TS did well and achieved 0.89 and 0.84 respectively.

In comparison our results for Hindi is more or less same as the BITS-Lipyantaran team achieved but for Bengali our results are significantly higher than the two best performing teams IITP-TS and JU-NLP-LAB.

6. Conclusion and Future Work

In this paper we discussed on generic back transliteration problem from romanized Indian social media text to language specific scripts. We collected a code-mixed corpus, annotated it with word level language markings and transliterated to respective languages. Tried a few existing models for on the dataset, but as mentioned making a comprehensive transliteration model for Indian SMT has various others challenges to meet, mostly context dependent homonyms, which is unattended in this paper.

This is an ongoing task. In future we would like to devote our time on context specific transliteration problem and would like to explore few more languages and with larger dataset.

References

Das, A., Ekbal, A., 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, Pages 80-83, August, 2009.

Gupta, K., Choudhury, M. and Bali, K., Mining Hindi-English Transliteration Pairs from Online Hindi Lyrics, In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), 2012. http://cse.iitkgp.ac.in/resgrp/cnerg/qa/fire13translit/index.html

Gupta, P., Bali, K., Banchs, R, Choudhury, M., Rosso, P. Query Expansion for Mixed-script Information Retrieval, in Proceedings of SIGIR 2014.

Karimi, K., Scholer, F., and Turpin, A. Machine Transliteration Survey. In ACM Computing Surveys (CSUR), Volume 43 Issue 3, April 2011.

Knight, K., Graehl, J.: Machine transliteration. Computational Linguistics 24(4) (1998), 599–612.

Kumaran, A., Khapra, M., and Li, H., Report of NEWS 2010 Transliteration Mining Shared Task, in the ACL 2010 Named Entities WorkShop (NEWS-2010), Uppsala, Sweden, Association for Computational Linguistics, July 2010.

Li, H., Kumaran, A., Pervouchine, V., and Zhang, M. Report of NEWS 2009 Machine Transliteration Shared Task, in the ACL/IJCNLP-2009 Named Entities WorkShop (NEWS-2009), Singapore, Singapore, Association for Computational Linguistics, August 2009.

Roy, R.S., Choudhury, M., Majumder, P., Agarwal, K. Overview and Datasets of FIRE 2013 Track on Transliterated Search. FIRE @ ISM. 2013.

Zhang, M., Li, H., Kumaran, A., and Liu, M. Report of NEWS 2012 Machine Transliteration Shared Task, in proceedings of the ACL 2012 Named Entities Workshop (NEWS), Jeju Island, South Korea, Association for Computational Linguistics, June 2012.