PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 108 JUNE 2017 319–329

Finite-State Back-Transliteration for Marathi

Vinit Ravishankar[a b]

[a] University of Malta, Faculty of Information and Communication Technology, Malta
[b] Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Czech Republic

Abstract

In this paper, we describe the creation of an open-source, finite-state based system for back-transliteration of Latin text in the Indian language Marathi. We outline the advantages of our system and compare it to other existing systems, evaluate its recall, and evaluate the coverage of an open-source morphological analyser on our back-transliterated corpus.

1. Introduction

Numerous transliteration standards that transliterate Indian languages from their various native scripts into the Latin script have existed for centuries, such as the Hunterian standard, or the International Alphabet of Sanskrit Transliteration (IAST). These standards are applied consistently within academic or formal contexts, where transliteration is necessary.

Despite the growth of Internet penetration in India, adoption of input method editors (IMEs) for Indian languages has been relatively slow. Monojit (2011) have provided a description of the potential challenges involved in creating IMEs with Latin-based keyboards. To compensate for this absence, there has been a tendency to represent Indian languages with the English variant of the Latin script on social media, forums, and over private messaging protocols like text messages or internet relay chat (IRC). Despite the existence of numerous formal transliteration standards, there is a strong tendency towards the use of an unofficial, "organic" transliteration standard, informally dubbed *Romanagari* (for languages that use the Devanagari script). Romanagari is largely based on the English variant of the Latin script, with no diacritics. (1) is an example of a Romanagari sentence, along with the formal ISO 15919 and Devanagari equivalents.

 Corresponding author: vinit.ravishankar@gmail.com
Cite as: Vinit Ravishankar. Finite-State Back-Transliteration for Marathi. The Prague Bulletin of Mathematical Linguistics No. 108, 2017, pp. 319–329. doi: 10.1515/pralin-2017-0030.

(1) mi tyanna marathi shikvaycho
mī tyāṁnā marāṭhī śikavāyacō
मी त्यांना मराठी शिकवायचो

"I used to teach them Marathi"

Being able to convert this text into Devanagari is fairly essential for any further processing, like machine translation, to even be attempted.

In this paper, we describe the implementation of a finite-state transducer (FST) based system to "back-transliterate" the southern Indo-Aryan language Marathi; i.e. to transliterate Romanagari Marathi to formal Devanagari. Whilst there has been no significant research on back-transliterating Marathi, we compare our results to back-transliteration results of other systems designed for Hindi, Bengali and Gujarati. We also outline the advantages of such a system over more statistical ones.

In section 2, we describe the language Marathi, including relevant grammatical details. Section 3 is a literature review of prior work in this domain. Section 4 describes the frameworks we used for implementing our system, and section 5 describes our methodology. Section 6 describes several interesting challenges we faced, and our (potential) solutions. In section 7, we describe our corpus, and our evaluation, and provide an analysis of the results. Finally, we discuss our results and our system in section 8, and conclude in section 9.

2. Marathi

Marathi, the fourth-most widely spoken language in India, is an Indo-Aryan language primarily spoken in the western Indian state of Maharashtra. Whilst historically, the Modi script was more widespread, modern Marathi is primarily written in the Balbodh script, which is an abugida, similar to Devanagari. Letters can be full vowels or full consonants; consonants are marked with diacritics to indicate associate vowels. The absence of a diacritic indicates a schwa, although not universally: Marathi, similar to other Indo-Aryan languages, displays the schwa deletion phenomenon (Choudhury et al., 2004), where inherent schwas associated with consonants are sometimes suppressed. Outside these environements, consonant clusters without schwas are often represented using ligatures, or with a "combining" diacritic (◌्).

Balbodh is very similar to Devanagari, apart from the addition of the retroflex lateral approximate (ळ), and an additional diacritic for consonant clusters beginning with an alveolar tap/trill in syllable onsets. Over the course of this paper, therefore, we refer to the script as Devanagari.

Grammatically, Marathi is more agglutinating than many other Indo-Aryan languages, likely owing to Maharashtra's geographical proximity to the Dravidian lan-

guage family: postpositions and cases are often orthographically joined to their heads, and enclitics are common. For instance, compare (2) and (3):

(2) Hindi:

baiṭh-n-ē vālē kō hī
sit-GER-OBL AGT to FOC

(3) Marathi:

bas-ṇār-yā-lā-c
sit-AGT-OBL-DAT-FOC

'To the person that is sitting *(and no one else)*'

3. Prior work

Research on south Asian[1] languages within the context of social media is quite abundant, with several studies focusing on code-switching, a fairly common phenomenon in the subcontinent. Transliteration — or, more accurately, *back*-transliteration — has been less of a research focus. A shared task that involved, amongst other challenges, back-transliteration of Hindi, Gujarati and Bengali was run in 2013 (Roy et al., 2013). The best-performing system (Gella et al., 2013) attempted to back-transliterate test using multi-view hashing.

Outside the south Asian context, there has been significant research on back transliteration. Knight and Graehl (1998) present an algorithm using weighted finite-state transducers, applied to Japanese; Kang and Choi (2000) present a decision tree-based system for Korean. Most of these systems, however, describe back-transliteration *to* the Roman script, rather than away from it.

There has also been some research (albeit not significant amounts) on the actual utility of using Romanagari: Rao et al. (2013) attempted to quantify the cognitive load of processing Romanagari Hindi and concluded that it was significantly higher than the load of processing both Devanagari Hindi and English. This has not, however, hindered the proliferation of Romanagari over social media.

4. Implementation

4.1. *hfst*

The Helsinki Finite-State Technology (*hfst*) library (Lindén et al., 2011) is a front-end for various open source finite-state library back-ends. It allows for data exchange

[1] "South Asian" in this context refers to languages spoken in the Indian subcontinent, including India, Pakistan, Bangladesh and Sri Lanka

between finite state tools implemented in multiple different formalisms; relevant to us, it covers the Xerox LexC and TwolC formalisms (Lindén et al., 2009). We used *hfst* to implement two-level rules; in our approach, this helped eliminate several problematic transcriptions that arose after mere orthographic transfer. Whilst traditionally used for morphological analysis, we view our problem in a very similar fashion: we obtain a set of multiple back-transliterations ("analysis") from which we choose the appropriate one ("disambiguation").

4.2. *lttoolbox*

For actual morphological analysis, we use the *lttoolbox* formalism, a morphological analysis framework used within the open-source machine translation framework, Apertium (Forcada et al., 2011). There exists an *lttoolbox*-based morphological analyser for Marathi[2], with a coverage of 80% on the Marathi Wikipedia. We measure the coverage of this analyser on the Devanagari generated by our system. Morphological analysis is an important prerequisite to many NLP tasks, including rule-based machine translation; measuring coverage of an open-source analyser is, therefore, a useful metric.

5. Methodology

As Romanagari → Devangari back-transliteration can be considered a many-to-many mapping, purely finite-state methods are not sufficient for transliteration, as our *hfst* output is a set of possible transliterations. To quote Knight and Graehl (1998), however, back-transliteration is less "forgiving" than transliteration: there can only be one correct equivalent to a word. We therefore compare three further "filters" to prune these lists: a frequency list, a 2-gram language model, and a morphological analyser.

Our *hfst* rules consisted of two layers: the first being a paradigm-based mapping from Devanagari characters to Latin; this layer relied on finite-state transducers to enforce appropriate transliteration in several domains, based on empirical rules we determined through observation, like inserting combining diacritics when moving from a consonant to another consonant. Our second layer consisted of a series of *twol*-style replace rules, applied synchronically. The most important rule here was to fix schwa deletion, i.e. to remove the combining diacritic wherever necessary. Other rules included, for instance, rules replacing certain digraphs with nasalisation diacritics.

6. Challenges

Whilst building this system, we faced several challenges that were rather interesting from a linguistic perspective.

[2]Available at `https://svn.code.sf.net/p/apertium/svn/languages/apertium-mar/`

6.1. Copular cliticisation

Spoken Marathi displays significant cliticisation of the copular verb असणे *asṇe* ("to be") in the present tense. These clitics, interestingly, carry more inflectional information than the formal copular verb would. For instance, contrast (4) and (5):

(4) tī basat āhē
3FSG sit-PTCP COP.3SG
"She is sitting"

(5) tī bast=iyē
3FSG sit-PTCP=COP.3FSG
"She is sitting"

This phenomenon is represented orthographically in Romanagari, where it is significantly more widespread than non-cliticised copulas are. Whether this cliticisation is valid from a prescriptive perspective in *formal* Devanagari or not is debatable; most formal grammars, such as Dhongade and Wali (2009) do not address this question, despite providing glosses (in Latin transcription) that include cliticised copulas. A brief analysis of several corpora with formal language seems to indicate that these forms are extremely infrequent; therefore, our default solution is to separate the copula from the participle. Our system does, however, allow this separation to be suppressed, based on relevant command-line parameters.

6.2. Word-final *a*

Unlike Hindi, Marathi does not always suppress word-final schwas, particularly for words with Sanskrit etymologies. Schwas are represented with the letter *a* in Romanagari. Further, masculine agreement for verbs and adjectives, often represented by the vowel आ (ā), is also represented with the letter *a*. Finally, neuter agreement - whilst represented with the letter *e* in formal contexts - is often reduced to a schwa in both spoken Marathi, marked with a nasalisation diacritic in Devanagari, and represented with the letter *a* in Romanagari. This leads, essentially, to a three-way back-transliteration ambiguity with word-final *a*s. This sort of ambiguity is impossible for a frequency-list based model to deal with; theoretically, our bigram model ought to be able to fix the agreement issues.

7. Evaluation

A serious problem with evaluating our system was the complete lack of corpora; previous shared tasks on similar themes did not include corpora in Marathi. To fix this, we created our own corpus for evaluating our analyser: a combination of three "mini" corpora (described in table 1), including:

1. Sections of the Marathi Wikipedia transliterated to Romanagari by three annotators[3]
2. Romanagari Twitter feeds (primarily viral "memes"), manually transliterated to Devanagari
3. Romanagari lyrics to Marathi songs, available on the internet, along with their formal Devanagari equivalents

Corpus	Tokens	Types	Letters
Wikipedia	666	450	12,922
Twitter	440	307	6,080
Lyrics	352	180	4,914
Total	1458	889	23,916

Table 1. Corpus statistics

We also generated our frequency list and language models using a Marathi corpus provided by the university, IIT Bombay[4] (3.8m tokens). We did not use Wikipedia to generate language models, as part of our evaluation was on Wikipedia-based text. We used the open-source *kenlm* (Heafield, 2011) to generate our bigram model.

Having assembled our corpus, we proceeded to evaluate our systems on it. Whilst Gella et al. (2013) evaluated their system using the F_1-score, this was not really a valid measure for our system: their precision metric measured the number of correct transliterations their system generated, divided by the number of transliterations their system generated, whilst our system was guaranteed to generate only one transliteration per word. A more valid metric for comparison would be the *recall* of our systems, which measured the number of correct transliterations generated, divided by the number of reference transliterations. We used this measure, along with the mean ambiguity: the mean number of candidates per word, generated by *hfst* before filtering.

7.1. Quantitative

We manually tokenised and lower-cased all our text, both Romanagari and Devanagari. Punctuation was stripped from both. We then ran our *hfst* system on our corpus and post-processed the output. For post-processing, we compared three models: unigram, bigram, and unigram with substring backoff, where we backed off to substrings until we received a match in our frequency list.

[3]All urban Maharashtrian native speakers of standard Marathi

[4]Available at `http://www.cfilt.iitb.ac.in/marathi_Corpus/`

	uni	bi	uni w/ substr
Recall (%)	68.74 (74.91)	68.10 (74.84)	70.37 (76.28)
Coverage (%)	72.72	71.83	72.65
Mean ambiguity	42.42		

Table 2. Evaluation with three filtering methods (figures in parentheses indicate recall on tokens)

Corpus	Recall (%)	Coverage (%)
Wikipedia	75.27	75.35
Twitter	69.58	76.36
Lyrics	64.28	70.82

Table 3. Per-corpus evaluation

For each, we measured recall and morphological analyser coverage. Our results have been described in table 2. A more fine-grained evaluation on each sub-corpus for our unigram with substring backoff model is described in table 3.

An interesting problem that we faced was the ambiguity/recall trade-off on adding or removing certain rules. The most obvious example of this was the schwa addition rule: despite the rule covering a significant chunk of relevant terms, there were exceptions to the rule, and issues with applying the rule at morpheme boundaries. Removing the rule, obviously, resulted in a significant drop in recall. Forcing the system to generate *both* possible forms for every valid context, however, increased the recall of our system. The mean ambiguity of our system, however, simultaneously increased massively. Two other rules include forcing long vowels at word-final positions, and short vowels at word-initial positions, and the inclusion of two non-native vowels used primarily in loanwords. The effects of the addition/removal of several such rules are outlined in table 4, with our unigram with substring matching model as the baseline.

Our evaluation shows that our system performs better than any of the systems outlined by Roy et al. (2013); their best systems obtained transliteration recalls of 50.90% and 47.50% for Bengali and Gujarati respectively, even ignoring proper nouns in their evaluation. Whilst obviously not directly comparable to Marathi, the (relative) linguistic similarity between Gujarati and Marathi provides at least some grounds for comparison.

	Baseline	Schwa	Vowel length	Foreign vowel
Recall (%)	70.37	72.57	70.15	70.37
Mean ambiguity (%)	42.42	250.25	40.12	169.48

Table 4. Changes in recall and ambiguity based on the inclusion of certain constraints (unigram w/ substr matching)

7.2. Qualitative

Interestingly, our results show no significant difference between the bigram model and the unigram one; the bigram model had the same gender agreement errors that the unigram ones did. A plausible explanation for this is that the frequency of specific determiner and noun combinations was low enough to be offset by the more frequent determiner, multiplied by a non-zero number after smoothing. We also evaluated models with higher-order *n*-grams; these showed no improvement.

We performed an analysis of 100 randomly sampled errors (table 5, page 327) from our set of back-transliteration failures. There were several interesting observations. First, most failures were foreign-language words. While many of these were English loanwords that were spelt the same as in English, and not phonetically (eg. *friend*, *perfume*), there were also several proper nouns from other Indian languages (primarily Hindi). Ambiguous back-transliterations were also an issue: often, multiple Devanagari words could be represented by a single Romanagari representation. This issue was particularly visible in agreement, and in word-final ambiguities between schwas and the vowel /a:/, where resolving a word-final *a* to either option would result in a valid word. Post-evaluation, we realised that it was possible to prevent some of this ambiguity: we introduced a post-processing measure that checked whether eliminating the word-final long vowel would result in a valid word. If it did, we included the long vowel. The justification for this was that whilst a word-final *a* could either represent a schwa or a long vowel, it would likely represent a long vowel where necessary to resolve ambiguity.

Our next source of failures was our schwa rule. These were of two different kinds: failures because the schwa rule inserted a schwa where none was necessary, and failures where it failed to insert a schwa where necessary. This was closely followed by words absent from our frequency list corpus. Next, we had 5 errors due to "impossible" to generate words, where the Romanagari was completely lossy: a character present in the source Devanagari was absent in its transliteration (eg. जेंव्हा *jeṃvhā* ("when"): Rom. *jevha*). Finally, two errors were due to problems with our rules, which we proceeded to fix post-evaluation.

Error type	Count
Foreign	30
Ambiguous input	29
Schwa rule fail	18
Unseen in corpus	16
"Incomplete" input	5
Rule absent	2

Table 5. Error analysis

8. Discussion

8.1. Analysis

The biggest issue with our system, at the moment, is our evaluation corpus itself. Whilst we did manage to assemble a reasonably diverse corpus, a larger corpus would allow for much better evaluation. More rigorous annotation control that accounted for variances in annotation style would also come in useful; Choudhury et al. (2010) describe several corpus bootstrapping techniques that we could use for more extensive future studies.

The most significant advantage of our finite-state system over statistical ones is the ability to easily model exceptions without having to retrain models. This makes it extremely trivial for us to add foreign words and proper nouns into our system. Further, due to the paradigm-oriented nature of *hfst*, adding the root of an OOV term would immediately allow for all concatenatively inflected forms of the noun to be (potentially) recognised. Setting up a finite-state system for back-transliteration, given linguistic knowledge - or even just native speaker intuition and a good grammar book - is also quite effortless. Based on our results, comparing similar systems for other south Asian languages, particularly parallel corpora-sparse ones, would be an interesting future project.

8.2. Improvements

Several of our ambiguity-related problems could be solved by taking context into account. Our bigram model was, unfortunately, not very successful: it would, however, be possible to integrate a morphological analyser into our pipeline. It could, for instance, determine the gender of the nearest noun and then force agreement on adjectives and verbs with ambiguous endings.

Whilst we currently rely on probabilities of complete generated strings, we could also integrated weighted FSTs into our system, where transitions are given certain weights based on their probabilities. Pereira and Riley (1997) have proposed a similar

system for speech recognition; Knight and Graehl (1998) adapted it to back-transliteration of Japanese katakana, implementing shortest-path algorithms to extract the most likely sequence.

9. Conclusions

There are several issues with schwas that need to be sorted out for our system to be truly "deployment" quality. However, it is important, again, to stress how "easy" it is to model exceptions with a finite-state based system. When attempting to gather social media text for analysis, our system could be used fairly trivially to obtain Devanagari equivalents to surface forms, which could then be rapidly post-edited. Common exceptions or loanwords could be obtained from a frequency list and added to the model prior to conversion.

Our system's recall on tokens - 76.28%, as described in table 2 on page 325 - is higher than our recall on terms, indicating that our system does perform better on more frequent words. This is quite encouraging, from a perspective of social media, where Romanagari is most likely to occur. Finally, our system is free and open-source, licensed under the GPL v3.0. This makes adoption for other Indian languages quite trivial, including for languages not included amongst the 22 "scheduled" languages of India, that are consequently (relatively) more underfunded and understudied.

Acknowledgements

We would like to thank Vaijayanti Ravishankar, Yamini Chitale and Anuja Phadke for their help with back-transliteration, to create part of our evaluation corpus. We also thank Francis Tyers for his comments on an earlier draft of this manuscript, Tommi Pirinen for his assistance with *hfst*, and the anonymous reviewers, whose insights and suggestions have been taken into account. This project has been funded by a stipend from the Erasmus Mundus Language and Communication Technology program, whom we are grateful to.

Bibliography

Choudhury, Monojit, Anupam Basu, and Sudeshna Sarkar. A diachronic approach for schwa deletion in Indo Aryan languages. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, pages 20–26. Association for Computational Linguistics, 2004.

Choudhury, Monojit, Kalika Bali, Tirthankar Dasgupta, and Anupam Basu. Resource creation for training and testing of transliteration systems for indian languages. 2010.

Dhongade, R. and K. Wali. *Marathi*. London Oriental and African language library. John Benjamins Publishing Company, 2009. ISBN 9789027238139. URL `https://books.google.com.mt/books?id=zVV0vi5C8uIC`.

Forcada, Mikel L., Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144, June 2011. ISSN 0922-6567, 1573-0573. doi: 10.1007/s10590-011-9090-0. URL http://link.springer.com/10.1007/s10590-011-9090-0.

Gella, Spandana, Jatin Sharma, and Kalika Bali. Query word labeling and back transliteration for indian languages: Shared task system description. *FIRE Working Notes*, 3, 2013.

Heafield, Kenneth. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics, 2011.

Kang, Byung-Ju and Key-Sun Choi. Automatic Transliteration and Back-transliteration by Decision Tree Learning. Citeseer, 2000.

Knight, Kevin and Jonathan Graehl. Machine transliteration. *Computational Linguistics*, 24(4): 599–612, 1998.

Lindén, Krister, Miikka Silfverberg, and Tommi Pirinen. HFST tools for morphology–an efficient open-source package for construction of morphological analyzers. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 28–47. Springer, 2009.

Lindén, Krister, Erik Axelson, Sam Hardwick, Tommi A Pirinen, and Miikka Silfverberg. Hfst—framework for compiling and applying morphologies. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 67–85. Springer, 2011.

Monojit, Umair Z Ahmed Kalika Bali. Challenges in designing input method editors for Indian languages: The role of word-origin and context. *Advances in Text Input Methods (WTIM 2011)*, page 1, 2011.

Pereira, FC and Michael D Riley. 15 Speech Recognition by Composition of Weighted Finite Automata. *Finite-state language processing*, page 431, 1997.

Rao, Chaitra, Avantika Mathur, and Nandini C Singh. 'Cost in Transliteration': The neurocognitive processing of Romanized writing. *Brain and language*, 124(3):205–212, 2013.

Roy, Rishiraj Saha, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. Overview of the fire 2013 track on transliterated search. In *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, page 4. ACM, 2013.

Address for correspondence:
Vinit Ravishankar
vinit.ravishankar@gmail.com
Faculty of Information and Communication Technology,
University of Malta,
Msida MSD 2080,
MALTA