Brahmi-Net: A transliteration and script conversion system for languages of the Indian subcontinent

Anoop Kunchukuttan *
IIT Bombay
anoopk@cse.iitb.ac.in

Ratish Puduppully *†
IIIT Hyderabad
ratish.surendran
@research.iiit.ac.in

Pushpak Bhattacharyya
IIT Bombay
pb@cse.iitb.ac.in

Abstract

We present *Brahmi-Net* - an online system for transliteration and script conversion for all major Indian language pairs (306 pairs). The system covers 13 Indo-Aryan languages, 4 Dravidian languages and English. For training the transliteration systems, we mined parallel transliteration corpora from parallel translation corpora using an unsupervised method and trained statistical transliteration systems using the mined corpora. Languages which do not have parallel corpora are supported by transliteration through a bridge language. Our script conversion system supports conversion between all Brahmi-derived scripts as well as ITRANS romanization scheme. For this, we leverage co-ordinated Unicode ranges between Indic scripts and use an extended ITRANS encoding for transliterating between English and Indic scripts. The system also provides top-k transliterations and simultaneous transliteration into multiple output languages. We provide a Python as well as REST API to access these services. The API and the mined transliteration corpus are made available for research use under an open source license.

1 Introduction

The Indian subcontinent is home to some of the most widely spoken languages of the world. It is unique in terms of the large number of scripts used for writing these languages. Most of the these are *abugida* scripts derived from the Brahmi script. *Brahmi* is one of the oldest writing systems of the Indian subcontinent which can be dated to at least the 3rd century B.C.E. In addition, Arabic-derived and Roman scripts are also used for some languages. Given the diversity of languages and scripts, transliteration and script conversion are extremely important to enable effective communication.

The goal of **script conversion** is to represent the source script accurately in the target script, without loss of phonetic information. It is useful for exactly reading manuscripts, signboards, etc. It can serve as a useful tool for linguists, NLP researchers, etc. whose research is multilingual in nature. Script conversion enables reading text written in foreign scripts accurately in a user's native script. On the other hand, **transliteration** aims to conform to the phonology of the target language, while being close to the source language phonetics. Transliteration is needed for phonetic input systems, cross-lingual information retrieval, question-answering, machine translation and other cross-lingual applications.

Brahmi-Net is a general purpose transliteration and script conversion system that aims to provide solutions for South Asian scripts and languages. While transliteration and script conversion are challenging given the scale and diversity, we leverage the commonality in the phonetics and the scriptural systems of these languages. The major features of *Brahmi-Net* are:

1. It supports 18 languages and 306 language pairs for statistical transliteration. The supported languages cover 13 Indo-Aryan language (Assamese, Bengali, Gujarati, Hindi, Konkani, Marathi, Nepali, Odia, Punjabi, Sanskrit, Sindhi, Sinhala, Urdu) , 4 Dravidian lan-

*These authors contributed equally to this project
†Work done while the author was at IIT Bombay

guages (Kannada, Malayalam, Tamil, Telugu) and English. To the best of our knowledge, no other system covers as many languages and scripts.

2. It supports script conversion among the following 10 scripts used by major Indo-Aryan and Dravidian languages: Bengali, Gujarati, Kannada, Malayalam, Odia, Punjabi, Devanagari, Sinhala, Tamil and Telugu. Some of these scripts are used for writing multiple languages. Devanagari is used for writing Hindi, Sanskrit, Marathi, Nepali, Konkani and Sindhi. The Bengali script is also used for writing Assamese. Also, Sanskrit has historically been written in many of the above mentioned scripts.

3. The system also supports an extended ITRANS transliteration scheme for romanization of the Indic scripts.

4. The transliteration and script conversion systems are accessible via an online portal. Some additional features include the ability to simultaneously view transliterations to all available languages and the top-k best transliterations.

5. An Application Programming Interface (API) is available as a Python package and a REST interface for easy integration of the transliteration and script conversion systems into other applications requiring transliteration services.

6. As part of the project, parallel transliteration corpora has been mined for transliteration between 110 languages pairs for the following 11 languages: Bengali, Gujarati, Hindi, Konkani, Marathi, Punjabi, Urdu, Malayalam, Tamil, Telugu and English. The parallel transliteration corpora is comprised of 1,694,576 word pairs across all language pairs, which is roughly 15,000 mined pairs per language pair.

2 Script Conversion

Our script conversion engine contains two rule-based systems: one for script conversion amongst scripts of the Brahmi family, and the other for romanization of Brahmi scripts.

2.1 Among scripts of the Brahmi family

Each Brahmi-derived Indian language script has been allocated a distinct codepoint range in the Unicode standard. These scripts have a similar character inventory, but different glyphs. Hence, the first 85 characters in each Unicode block are in the same order and position, on a script by script basis. Our script conversion method simply maps the codepoints between the two scripts.

The Tamil script is different from other scripts since it uses the characters for unvoiced, unaspirated plosives for representing voiced and/or aspirated plosives. When converting into the Tamil script, we substitute all voiced and/or aspirated plosives by the corresponding unvoiced, unaspirated plosive in the Tamil script. For Sinhala, we do an explicit mapping between the characters since the Unicode ranges are not coordinated.

This simple script conversion scheme accounts for a vast majority of the characters. However, there are some characters which do not have equivalents in other scripts, an issue we have not addressed so far. For instance, the Dravidian scripts do not have the *nukta* character.

2.2 Between a Roman transliteration scheme and scripts from the Brahmi family

We chose ITRANS[1] as our transliteration scheme since: (i) it can be entered using Roman keyboard characters, (ii) the Roman character mappings map to Indic script characters in a phonetically intuitive fashion. The official ITRANS specification is limited to the Devanagari script. We have added a few extensions to account for some characters not found in non-Devanagari scripts. Our extended encoding is backward compatible with ITRANS. We convert Devanagari to ITRANS using Alan Little's python module[2]. For romanization of other scripts, we use Devanagari as a pivot script and use the inter-Brahmi script converter mentioned in Section 2.1.

3 Transliteration

Though Indian language scripts are phonetic and largely unambiguous, script conversion is not a sub-

[1] http://www.aczoom.com/itrans/

[2] http://www.alanlittle.org/projects/transliterator/transliterator.html

stitute for transliteration which needs to account for the target language phonology and orthographic conventions. The main challenges that machine transliteration systems encounter are: script specifications, missing sounds, transliteration variants, language of origin, etc. (Karimi et al., 2011). A summary of the challenges specific to Indian languages is described by Antony, P. J. and Soman, K.P. (2011).

3.1 Transliteration Mining

Statistical transliteration can address these challenges by learning transliteration divergences from a parallel transliteration corpus. For most Indian language pairs, parallel transliteration corpora are not publicly available. Hence, we mine transliteration corpora for 110 language pairs from the ILCI corpus, a parallel translation corpora of 11 Indian languages (Jha, 2012). Transliteration pairs are mined using the unsupervised approach proposed by Sajjad et al. (2012) and implemented in the *Moses* SMT system (Durrani et al., 2014). Their approach models parallel translation corpus generation as a generative process comprising an interpolation of a transliteration and a non-transliteration process. The parameters of the generative process are learnt using the EM procedure, followed by extraction of transliteration pairs from the parallel corpora.

Table 1 shows the statistics of mined pairs. We mined a total of 1.69 million word pairs for 110 language pairs. We observed disparity in the counts of mined transliteration pairs across languages. Language pairs of the Indo-Aryan family from geographically contiguous regions have more number of mined pairs. For instance, the *hin-pan, hin-guj, mar-guj, kok-mar* pairs have high number of mined transliterations averaging more than 30,000 entries. The mined pairs are diverse, containing spelling variations, orthographic variations, sound shifts, cognates and loan words.

3.2 Training transliteration systems

We model the transliteration problem as a phrase based translation problem, a common approach which learns mappings from character sequences in the source language to the target language. The systems were trained on the mined transliteration parallel corpus using *Moses*. The mined pairs are first segmented and a phrase-based machine translation system is trained on them.

We used a hybrid approach for transliteration involving languages for which we could not mine a parallel transliteration corpus. Source languages which cannot be statistically transliterated are first transliterated into a phonetically close language (bridge language) using the above-mentioned rule-based system. The bridge language is then transliterated into the target language using statistical transliteration. Similarly, for target languages which cannot be statistically transliterated, the source is first statistically transliterated into a phonetically close language, followed by rule-based transliteration into the target language.

4 *Brahmi-Net* Interface

Brahmi-Net is accessible via a web interface as well an API. We describe these interfaces in this section.

4.1 Web Interface

The purpose of the Web interface is to allow users quick access to transliteration and script conversion services. They can also choose to see the transliteration/script conversion output in all target languages, making comparison easier. Alternative choices of transliteration can also be studied by requesting the top-5 transliterations for each input. A snapshot of the interface is shown in Figure 1. The web interface is accessible at:
`http://www.cfilt.iitb.ac.in/brahminet/`

4.2 REST API

We provide a REST interface to access the transliteration and script conversion services. Simultaneous transliterations/script conversion into all languages and top-k transliterations are also available. The REST endpoints have an intuitive signature. For instance, to fetch the transliteration for a word from English (en) to Hindi (hi), the REST endpoint is:

`http://www.cfilt.iitb.ac.in/indicnlpweb/indicnlpws/transliterate/en/hi/<input>/statistical`

The API returns a serialized JSON object containing a dictionary of target language to top-k transliterations. The detailed API reference is available on the website.

	hin	urd	pan	ben	guj	mar	kok	tam	tel	mal	eng
hin	-	21185	40456	26880	29554	13694	16608	9410	17607	10519	10518
urd	21184	-	23205	11379	14939	9433	9811	4102	5603	3653	5664
pan	40459	23247	-	25242	29434	21495	21077	7628	15484	8324	8754
ben	26853	11436	25156	-	33125	26947	26694	10418	18303	11293	7543
guj	29550	15019	29434	33166	-	39633	35747	12085	22181	11195	6550
mar	13677	9523	21490	27004	39653	-	31557	10164	18378	9758	4878
kok	16613	9865	21065	26748	35768	31556	-	9849	17599	9287	5560
tam	9421	4132	7668	10471	12107	10148	9838	-	12138	10931	3500
tel	17649	5680	15598	18375	22227	18382	17409	12146	-	12314	4433
mal	10584	3727	8406	11375	11249	9788	9333	10926	12369	-	3070
eng	10513	5609	8751	7567	6537	4857	5521	3549	4371	3039	-

Table 1: Mined Pairs Statistics (ISO-639-2 language codes are shown)

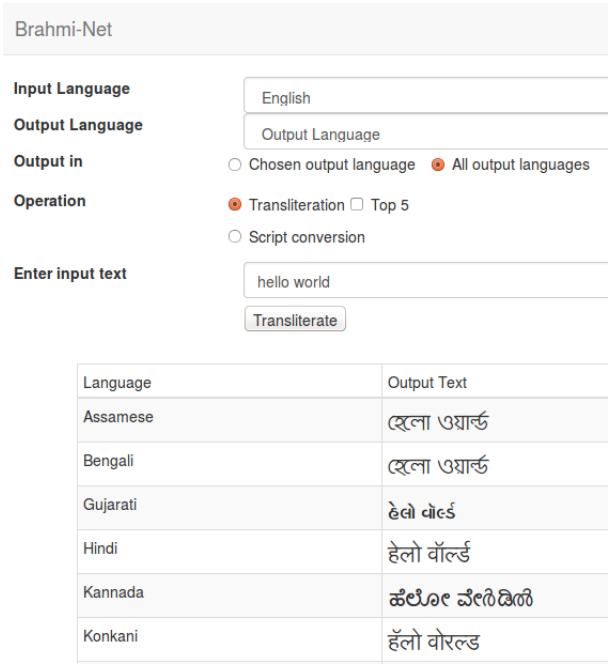

Figure 1: *Brahmi-Net* Web Interface

Lang Pair	Rule	Statistical	
		top-1	*top-5*
ben-mar	64.6	68.3	87.1
mal-tam	27.9	30.9	66.0
mar-ben	68.0	67.3	85.2
tel-mar	68.2	70.9	87.5

Table 2: Transliteration Accuracy (%)

5 Evaluation

5.1 Transliteration Accuracy

We evaluated the top-1 and top-5 transliteration accuracy for a sample set of language pairs. For this evaluation, we used an internally available, manually created corpus of 1000 transliteration pairs for each language pair. These transliterations were manually curated from synsets in *IndoWordNet*[3] Though this corpus does not reflect the diversity in the mined transliterations, evaluation on this corpus could be a pointer to utility of the transliteration corpus. We compare the accuracy of match for transliteration against the rule based script conversion output for some language pairs. Table 2 shows the accuracy values. *top-1* indicates exact match for the first transliteration output returned by our system, whereas *top-5* indicates match in the top 5 transliterations returned by the system.

5.2 Case Study: Improving SMT output

Our work in developing the transliteration systems was initially motivated by the need for transliterating the untranslated words in SMT output. To evaluate the transliteration systems in the context of machine translation, we post-edited the phrase based system (PB-SMT) outputs of Indian language pairs provided by Kunchukuttan et al. (2014) using our transliteration systems. Each untranslated word was replaced by each of its top-1000 transliterations and the resulting candidate sentences were re-ranked using a language model. We observe a significant improvement in translation quality across language pairs, as measured by the BLEU evaluation metric. Due to space constraints, we present results for only 8 language pairs in Table 3. We observed that though the system's best transliteration is not always correct, the sentence context and the language model select the right transliteration from the top-k transliteration

[3] http://www.cfilt.iitb.ac.in/indowordnet

Lang Pair	PB-SMT	PB-SMT +translit
urd-eng	21.0	21.59
tel-eng	12.09	12.34
kok-ben	24.61	27.69
pan-hin	71.26	75.25
mar-pan	34.75	36.92
tel-mal	6.58	7.54
guj-tel	16.57	18.61
tal-urd	15.65	16.22

Table 3: Results of PB-SMT output + transliteration of OOVs (%BLEU)

candidates. The top-k transliterations can thus be disambiguated by SMT or other downstream applications.

6 Conclusion

Brahmi-Net is an effort to provide a comprehensive transliteration and script conversion solution for all languages of the Indian subcontinent. Unsupervised transliteration mining and leveraging the phonetic and scriptural similarities between the languages have been the key ingredients in scaling the system to a large number of languages. Even the simple phrase based SMT model of transliteration has proved useful for transliterating the output of MT systems. A natural extension would be to employ richer transliteration models. There is scope for improvement in the hybrid models of transliteration used in the system. Some of the finer details regarding script conversions have to be ironed out. Finally, a long term goal is to support other major languages from South Asia, which differ phonetically from the Indo-Aryan and Dravidian languages or use non-Brahmi scripts.

Acknowledgments

We would like to thank Arjun Atreya for making available parallel transliterations from *IndoWordNet* for evaluation of our system.

References

Antony, P. J. and Soman, K.P. 2011. Machine Transliteration for Indian Languages: A Literature Survey. *International Journal of Scientific and Engineering Research*.

Nadir Durrani, Hieu Hoang, Philipp Koehn, and Hassan Sajjad. 2014. Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. *EACL 2014*.

Girish Nath Jha. 2012. The TDIL program and the Indian Language Corpora Initiative. In *Language Resources and Evaluation Conference*.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2011. Machine transliteration survey. *ACM Computing Surveys*.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee, Ritesh Shah, and Pushpak Bhattacharyya. 2014. Sata-Anuvadak: Tackling Multiway Translation of Indian Languages. In *Language Resources and Evaluation Conference*.

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.