

BY PUSHPAK BHATTACHARYYA, HEMA MURTHY,  
SURANGIKA RANATHUNGA, AND RANJIVA MUNASINGHE

# Indic Language Computing

IN APRIL 2019, following the Easter Sunday bomb attacks, the Government of Sri Lanka had to shut down Facebook and YouTube for nine days to stop the spreading of hate speech and false news, posted mainly in the local languages Sinhala and Tamil. This came about simply because these social media platforms did not have the capability to detect and warn about the provocative content.

India's Ministry of Human Resource Development (MHRD) wants lectures on Swayam<sup>a</sup> and NPTEL<sup>b</sup>—the online teaching platforms—to be translated into all Indian languages. Approximately 2.5 million students use the Swayam lectures on computer science alone. The lectures are in English, which students find difficult to understand. A large number of lectures are manually subtitled in English. Automatic speech recognition and machine translation into Indian languages will be great enablers for the marginalized sections of society.

Requirements like these are real and abundant.

a <https://swayam.gov.in/>

b <https://nptel.ac.in/>

These are social and commercial needs, whose servicing requires user interaction and information dissemination in languages other than English. Only around 10% of India's population, or about 125 million people, can speak English; only about half that number is comfortable reading and writing in that language. The social media activity of the youth of the Indian subcontinent (where 65% of the population is below the age of 35) generates a huge amount of e-content, much of which is in text form, is multilingual, and even code-mixed (text in multiple languages at the same time, often in Roman script). The numbers are mind-boggling:<sup>c</sup>

- ▶ 462.1 million Internet users (34% of the population; the global average is 53%).
- ▶ 430.3 million users access the Internet via mobile devices (79% of total Web traffic).
- ▶ 250 million social media users (19% of the population; the global average is 42%).
- ▶ 260 million WhatsApp users, and 53 million Instagram users.

Sri Lanka alone has seven million Internet users (2018 data), which equates to a penetration of 32%.

There is no doubt that speech and natural language processing (NLP) of Indic languages is hugely important and relevant, and has the potential to influence the lives and activity of at least 20% of the world's population.

## Challenges of Indian Language Computing

The Indian subcontinent is divided into seven independent countries: India, Pakistan, Bangladesh, Nepal, Bhutan, Sri Lanka, and the Maldives.

There are approximately 1,599 languages in India, out of which about 420–440 are in active use. Languages in the region fall into four major linguistic groups: Indo-Aryan (spoken mainly in the northern part of south Asia and in Sri Lanka), Dravidian (spoken mainly in south India), Tibeto-Burman (spoken mainly in northeast India), and

c *India Today*, April 2018 issue.



Diversity is the name of the game for Indic-language computing; shown here are scripts in Devanagari, Brahmi, Odia, Tamil, Telugu, Malayalam, and Sinhala, among other languages.



Austro-Asiatic (Khasi in Meghalaya, and Munda in Chhotonagpur). These language families each have their own linguistic characteristics, whose richness and complexity have been delved into in multiple scholarly treatises.<sup>11</sup> These complexities, along with technohuman constraints, give rise to the challenges of Indic language computing, some of which are described here.

**Scale and diversity.** For Indic languages, solutions must be simultaneously proposed for multiple languages. There are 22 major languages in India, written in 13 different scripts, with over 720 dialects. There is a need to develop approaches that are generic, and scaling to multiple languages should be only a task of adaptation. As the languages are quite different, there is a lot of effort required to arrive at common solutions. Although E2E (end-to-end) is the buzzword today, use of multiple scripts for Indian languages makes systems complex (as illustrated in the accompanying figure).

**Long utterances.** Indian-language utterances are much longer in duration compared to English, and hardly contain punctuation. A typical English sentence has about 70 characters, while a sentence in an Indian language typically averages 130 characters. E2E systems perform poorly with long sentences.

**Code mixing.** Code mixing is the use of more than one language in text/utterance. Handling code switching from one language to another in both automatic speech recognition (ASR) and text to speech (TTS) is a challenge. In ASR, the

language boundary could be an important cue for semantics (assuming the lexicon accounts for the vocabulary of both languages). Also, Indian language words are included in an English sentence, where gerundification (such as “I’m chalaaoing a car,” meaning “I am driving a car”) of Indian-language nouns is common. In TTS, producing code-switched systems requires the prosodic characteristics of the language and the speaker are preserved, especially when code switching involves stress-timed and syllable-timed languages. The interplay between languages in terms of prosody needs to be understood to make the sentences sound natural.

**Resource scarcity.** Indic-language computing is bogged down by paucity of data. Language computing these days is primarily data-driven, with sophisticated machine learning techniques employed on the data. The success of these approaches depends crucially on the availability of large amounts of high-quality data. We take the example from automatic machine translation (MT), which is highly data-driven these days: the Hansard corpus for English-French contains 1.6 billion words; the Europarl Parallel Corpus for 21 European languages contains about 30 million words; WMT 15 data for English-Czech contains about 16 million parallel sentences; and WMT 14 data for English-German contains about 4.5 million parallel sentences. An Indic-language example with comparable size is the CFILT-IITB English-Hindi corpus, which includes 800,000 parallel sentences.

Other languages offer very little language data. For example, available parallel corpora for Sinhala-Tamil are well below 50,000 sentences. Even raw, clean corpora are of great value for language computing. Modern-day deep learning techniques start with word embeddings (WEs). WEs are learned from huge amounts of corpora (millions of words) that capture the context distribution for words and phrases. Such distribution captures semantics, which is an elusive entity, computationally speaking. Many Indic languages do not have a processable clean corpus from word lists, WEs, and a rich lexicon can be built. Another application area that is affected by paucity of data is ASR-TTS. Spoken signals must be correct, with proper text units. Then there are transcriptions of spoken utterances that need to be accurate. Although there are subtitled YouTube videos and lectures, they require curation, as time alignments are quite poor. However, the number of available hours of training data is small, leading to poor alignments.

**Absence of basic speech and NLP tools.** The NLP pipeline starts with word-level processing, and goes all the way up to discourse computation (connecting many sentences together with attention to coherence and cohesion).<sup>2</sup> The tools used at each stage of this pipeline are affected by the accuracy of tools in the preceding stages. For English, since many groups across the world have worked on the computational processing of the language, a staged development of NLP tools of English occurred. NLTK,<sup>d</sup> a GATE-like<sup>e</sup> NLP framework came into being, paving the way for large application development in English. In contrast, even basic morphology analyzers that split words into their roots and suffixes do not exist for most Indic languages, and even if they exist, their accuracy level is low.

**Absence of linguistics knowledge.** Though speech processing and NLP are data-driven, linguistics insight and understanding of language phenomena often help solve the problem of accuracy saturation. Deep understanding of language phenomena helps design

<sup>d</sup> <https://www.nltk.org/>

<sup>e</sup> <https://gate.ac.uk/>

good problem-solving strategies, and helps immensely in error analysis and explainability. Many Indic languages do not have a linguistics tradition.

**Script complexity and non-standard input mechanisms.** In an Indic language such as Devanagari, there are 13 vowels, 33 consonants, 12 vowel marks or matras, complex conjunct characters, and special symbols such as anusvara, visarga, chandra bindu, and Nukta.<sup>f</sup> This makes input speed slow (8–10 words per minute, compared to 20–30 w.p.m. in English). Though an InScript keyboard layout has been mandated by the Government of India, there are questions on its optimality and ease of use. Suggestions for more efficient keyboard layouts keep appearing. The problem is compounded by the presence of 13 different scripts, which drives people to resort to Roman input through transliteration most of the time.

**Non-standard transliteration.** There are variations in representation when it comes to transliteration in Roman. For example, the Hindi word for “mango” (a fruit) can be transliterated as “am,” “Am,” or “aam.” This creates a challenge for processing, and does not help the English-illiterate.

**Non-standard storage.** The appearance of Unicode for Indic languages and its adoption as the standard encoding of Indic language e-content was rather slow. As a result, many proprietary fonts exist, and the content of those fonts require downloading and algorithmic adaptation.

**Man-made problems.** Problems are further compounded by the fact that noise levels on the subcontinent average about 70dB, while the maximum permissible level is about 55dB. This challenges speech recognition technologies.

**Some challenging language phenomena.** A language phenomenon across major Indian languages is compound verbs (CVs), whose processing is a must for Indic-language NLP (INLP). CVs are composed of two verbs such that the main information content of actual action is carried by the first verb (called the polar) and the Gender-Number-Tense-Aspect-Modality (GNPTAM) information are marked on the second verb (called the vector). Elaborate machinery is needed for computational processing of

CVs, starting from morphology, and up to the pragmatic level.<sup>3</sup> As an illustration, consider the Hindi compound verb:<sup>g</sup>

H<sub>1</sub>: *bol uthaa* (Hindi string)

G<sub>1</sub>: *speak rose* (gloss)

T<sub>1</sub>: *spoke up* (English translation)

There is a sense of abruptness/urgency/letting-out-pent-up-feeling that is an additional layer of meaning carried by the vector verb on top of the main action of speaking (the polar). Catching such fine nuance is essential, for example, in sentiment and emotion analysis.<sup>8</sup>

**Morpheme stacking.** Many Indian languages show heavy stacking of morphemes (for the example, subscript 2 means the second sentence in the document):

M<sub>2</sub>: *gharaasamorच्याanii malaa saamgit* (Marathi sentence).

P<sub>2</sub>: *ghar+aa+samor+chyaa+nii+mala a+saMgit+le* (showing morphemes).

G<sub>2</sub>: *house+<morpheme: oblique marker>+front+of+<ergative marker: agent> me told* (gloss).

T<sub>2</sub>: The one in front of the house told me (translation).


This example is typical of the processing of most Indic languages. P<sub>2</sub> (denoting parts) shows the constituents of the word strings. This needs sophisticated word segmenters and morphology analyzers.

### State of the Art and Achievements


Despite the aforementioned challenges, the Indic language computing community has taken notable strides forward. This is seen on multiple fronts, such as corpus creation, NLP tool-building, end-user application development, research funding, collaboration, and standards and policy setting.

Fortunately for NLP, huge amounts of text in electronic form have become available in many walks of life (such as customer interactions in banks, reviews of online companies, judicial documents, contracts, e-books, and so on), paving the way for researchers to think about and apply powerful machine learning techniques to language technology problems. A case in point is the use of Europarl Parallel Corpus

<sup>g</sup> We use transliterated Roman script for universal readability: H<sub>1</sub>- sentence no. 1, which is in Hindi; G<sub>1</sub>- word for word translation of sentence no. 1 called gloss; T<sub>1</sub>- translation in English of sentence no 1.



**There is no doubt that speech and natural language processing of Indic languages is hugely important and relevant, and has the potential to influence the lives and activity of at least 20% of the world's population.**



<sup>f</sup> These are diacritic marks.

**The Si-Ta translation system was developed as a solution to the scarcity of Sinhala-Tamil translators in the government sector. The system has already shown better performance than the commonly used Google Translate for the selected domain.**

in creating automatic MT systems. A game-changer came in 2005, when 110 pairs of statistical machine translation (SMT) systems were created by applying machine learning on this resource,<sup>5</sup> ushering in the era of SMT. Another paradigm shift came in the form of neural machine translation (NMT) in 2014, beating SMT by a wide margin.<sup>1</sup> The lesson is obvious: feed language data to ML algorithms to create NLP systems.

One of the authors of this article replicated the SMT and NMT research on Indian languages with his research team and wound up with state-of-the-art results for translation involving Indian languages and English.<sup>6,9</sup> The data used for training was the ILCI corpora<sup>4</sup> created at the initiative of the Technology Development in Indian Languages (TDIL) program of the Ministry of Electronics and Information Technology (MEITY), along with the Indian Institute of Technology Bombay (IIT Bombay) parallel corpus<sup>8</sup> created at the Center for Indian Language Technology of IIT Bombay.<sup>h</sup>

There have also been some isolated efforts to develop NLP applications to cater to specific needs in the region. One example is the Si-Ta machine translation system developed for Sinhala-Tamil to be used by the government sector of Sri Lanka. This translation system was developed as a solution to the scarcity of Sinhala-Tamil translators in the government sector. Despite the small parallel corpus used, the system has already shown better performance than the commonly used Google Translate for the selected domain.<sup>10</sup>

TDIL-MEITY has provided great service to the cause of Indian language technology (ILT) development. Since 2000, TDIL has been instrumental in initiating, funding, and sustaining research and development in ILT, including unicode standard, scripts, input methods, speech (<http://www.iitm.ac.in/donlab/tts/>), optical character recognition (OCR), MT, and cross-lingual information retrieval in Indian languages.<sup>i</sup> These initiatives have produced know-how, tools, and resources (like Indian-language Wordnets<sup>j</sup>) that

h <http://www.cfilt.iitb.ac.in>

i Very informative articles on large consortia projects in ILT can be found at <http://tdil.meity.gov.in/Publications/Vishwabharatnew.aspx>.

j <http://www.cfilt.iitb.ac.in/indowordnet/>

are now ready to be commercialized through industry adoption and start-ups.

A recent initiative by NITI-Aayog,<sup>k</sup> the premier policy think tank of the Government of India, under the chairmanship of the Prime Minister of India providing both directional and policy inputs, brought together Indian academia, start-ups, industry, and research labs to discuss traction and monetization of ILT. It was decided to create an NLP access repository that would enable start-ups and industry to create large ILT applications, such as online review sentiment analyzers in Indian languages. The access repository will provide a platform from which to launch large applications.

The Bureau of Indian Standards of India's Ministry of Commerce recently set up a panel on Artificial Intelligence Standardization (LITD30).<sup>l</sup> This is the Indian mirror of SC 42, the sectional committee of the International Standards Organization (ISO) for AI standardization. Language Technology and its standardization is an important focus of LITD30, especially in the context of trustworthiness and certification (that is, automatic detection of fake news). Other noteworthy efforts on the subcontinent have been reported by the Language Technology Research Laboratory of Sri Lanka's University of Colombo,<sup>m</sup> the National Language Processing Centre of Sri Lanka's University of Moratuwa,<sup>n</sup> and the Center for Language Engineering<sup>o</sup> of Pakistan's Al-Khwarizmi Institute of Computer Science University of Engineering and Technology.

### Way Forward

We close this discussion with a few pointers for moving forward:

► Although languages are quite distinct, there are also a number of similarities, in that all the languages can be represented by a superset of sounds, which is much less than the number of graphemes that make up all the languages. A unified representation is the current need to enable speech

k <http://www.niti.gov.in/>

l <https://bis.gov.in/wp-content/uploads/2018/11/agenda-compo-litd-30.pdf>

m <http://ltrl.ucsc.lk/>

n <https://www.mrt.ac.lk/web/nlp>

o <http://www.cle.org.pk/>

and language technologies. This will help pool low resources across various languages to build robust ASR systems for Indian languages.

► In the context of TTS, the major issue to be addressed is the input method. Text is available in multiple Indian scripts, but digital resources in terms of high-quality parallel corpora are few and far between. In the context of both ASR and TTS, generic acoustic models across various languages, generic language models in the former, and a generic Indic voice in the latter need to be designed. This will also address the issue of code switching.

► In TTS, code mixing must find ways to preserve the speaker's voice across languages. Further, the influence of the native tongue on a non-native tongue must be preserved. For instance, there are as many varieties of English as there are native tongues. Replacing non-native English (which is syllable-timed) with stress-timed English can make it difficult for the listener to understand.

► Text in social media generally includes code switching/mixing. Further, there are many words that have a local cultural connotation. Building language resources to address these requires the expertise of linguists, speech scientists, natural language processing engineers, and ethnographers.

► Data is the new oil, and NLP and ILT is no exception. There is no doubt that resources with quality and coverage need to be created, and created fast. Thinking creatively on how to engage even a small portion of 1 billion hands for resource creation is a must. Crowdsourcing, in spite of its criticism with respect to quality, seems to be the way forward. Providing attractive, helpful interfaces and remuneration can go a long way toward resource creation. In this context, the Language Data Consortium for Indian Languages (LDC-IL)<sup>p</sup> initiative of Central Institute of Indian Languages (CIIL) is noteworthy.

► Evaluation is the key to actual use of language resources and should be taken very seriously. Like TREC<sup>q</sup> (USA), CLEF<sup>r</sup> (Europe), and NTCIR<sup>s</sup> (CJK countries), India's Forum for Information Retrieval

Evaluation (FIRE) initiative<sup>t</sup> has taken up the cause of evaluation in information retrieval and allied tasks. A FIRE-like initiative is needed for all areas of ILT.

### Conclusion

Indic Language Computing (ILC) is too important a problem to be lying in oblivion. Given spectacular advancements to date in computing science and technology, Internet, AI, machine learning, and NLP, the time is ripe for a concerted thrust for realization and social penetration of ILC. The energy of the start-up echo system has to be harnessed with government support, and guidance from academia. Language resource creation is a precondition for ILC revolution, and as in all cases of large infrastructure building (roads, internet, gas lines, waterways), government sponsorship is needed for resource building.

t <http://fire.irs.res.in/fire/2019/home>

### References

1. Bahdanau, D., Cho, K. and Bengio, Y. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.
2. Bhattacharyya, P. Natural language processing: A perspective from computation in presence of ambiguity, resource constraint and multilinguality. *CSI J. Computer Science and Engineering* 1, 2 (2012).
3. Chakrabarti, D., Mandalia, H., Priya, R., Sarma, V., and Bhattacharyya, P. Hindi compound verbs and their automatic extraction. In *Proceedings of Computational Linguistics*, Manchester, U.K., Aug. 2008.
4. Jha, G.N. The TDIL program and the Indian language corpora initiative. In *Proceedings of LREC*, 2010.
5. Koehn, P. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Machine Translation Summit*, 2005.
6. Kunchukuttan, A., Mishra, A., Chatterjee, R., Shah, R. and Bhattacharyya, P. Shata-Anuvadak: Tackling multiway translation of Indian languages. In *Proceedings of the Language Resources Evaluation Conference*, 2014.
7. Kunchukuttan, A., Mehta, P., and Bhattacharyya, P. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of LREC*, (Miyazaki, Japan, May 7–12, 2018).
8. Liu, B. *Sentiment Analysis and Opinion Mining*. Morgan and Claypool Publishers, 2012.
9. Murthy, R., Kunchukuttan, A., and Bhattacharyya, P. Addressing word-order divergence in multilingual neural machine translation for extremely low resource languages. In *Proceedings of LREC*, 2019.
10. Ranathunga, S., Farhath, F., Thayasivam, U., Jayasena, S., and Dias, G. Si-Ta: Machine translation of Sinhala and Tamil official documents. In *Proceedings of the National Information Technology Conference*, 2019.
11. Subbarao K.V. *South Asian Languages—A Syntactic Typology*. Cambridge, 2012.


**Pushpak Bhattacharyya** (pb@cse.iitb.ac.in) is a professor in the computer science and engineering department of IIT Bombay, and director of IIT Patna.

**Hema Murthy** (hema@cse.iitm.ac.in) is a professor in the computer science and engineering department of IIT Madras.


**Surangika Ranathunga** (surangika@cse.mrt.ac.lk) is a senior lecturer in the department of computer science and engineering and a member of the faculty of engineering at the University of Moratuwa.

**Ranjiva Munasinghe** (ranjiva@mindlanka.org) is chief executive officer of MIND Analytics and Management in Colombo, Sri Lanka.

© 2019 ACM 0001-0782/19/11 \$15.00



**Code mixing must find ways to preserve the speaker's voice across languages. Further, the influence of the native tongue on a non-native tongue must be preserved.**



p <http://www.ldcil.org/>

q <https://trec.nist.gov/>

r <http://www.clef-initiative.eu/>

s <http://research.nii.ac.jp/ntcir/index-en.html>