SandhiKosh: A Benchmark Corpus for Evaluating Sanskrit Sandhi Tools

Shubham Bhardwaj, Neelamadhav Gantayat, Nikhil Chaturvedi, Rahul Garg, Sumeet Agarwal
Indian Institute of Technology, IBM Research
New Delhi, INDIA
shubhamiitd.007@gmail.com, neelamadhavg@in.ibm.com, 709nikhil@gmail.com,
rahulgarg@cse.iitd.ac.in, sumeet@ee.iitd.ac.in

Abstract

Sanskrit is an ancient Indian language. Several important texts which are of interest to people all over the world today were written in Sanskrit. The Sanskrit grammar has a precise and complete specification given in the text Aṣṭādhyāyī by Pāṇini. This has led to the development of a number of *Sanskrit Computational Linguistics* tools for processing and analyzing Sanskrit texts. Unfortunately, there has been no effort to standardize and critically validate these tools. In this paper, we develop a Sanskrit benchmark called SandhiKosh to evaluate the completeness and accuracy of Sanskrit Sandhi tools. We present the results of this benchmark on three most prominent Sanskrit tools and demonstrate that these tools have substantial scope for improvement. This benchmark will be freely available to researchers worldwide and we hope it will help everyone working in this area evaluate and validate their tools.

1. Introduction

On the 11th of December, 2014, the United Nations General Assembly unanimously passed a resolution declaring the summer solstice, June 21st, as the International Day of Yoga (UN, 2014). The resolution recognizes that "yoga provides a holistic approach to health and well-being" and that "a wider dissemination of information about the benefits of practicing yoga would be beneficial for the health of the world population". The popularity of Yoga has been growing steadily all over the world primarily due to its potential health and healing benefits in ailments such as depression (Uebelacker et al., 2010), cardiovascular disorders (Raub, 2002) and other chronic diseases (Yang, 2007).

With the growing worldwide popularity of this ancient discipline, there is also a growing interest to understand and practice this discipline in its pure and unadulterated form. All of the classic texts of this discipline such as Gherand Samhita (Saraswati, 2013), Hath-Yoga-Pradiptika (Singh, 1915), Yoga-Vashistha (Mitra, 1891), Patanjali-Yoga-Sutra (Prasada, 1998) and Bhagavad-gītā (Swarupananda, 2016) were composed in the Sanskrit language. While translations of some of these texts are available in other languages, for a serious Yoga practitioner, reading and understanding these texts directly in their native language has a lot of value. Unfortunately, Sanskrit is now spoken by only a small number of people. These classical texts, although available, remain inaccessible to most of the world.

Due to this and some more reasons, there is growing interest in learning the Sanskrit language (HT, 2007; Ghosh, 2015). For example, the St. James Junior school in London has introduced Sanskrit language in the junior school because the "knowledge of grammar ultimately gives the pupils a greater clarity and accuracy in thinking, reading and speaking" (SJJS, 2017). Fortunately, the Sanskrit language has undergone very little modification and by learning this language, it is possible to read and understand most of the classical texts in Sanskrit, including those which date back to centuries BC.

One of the major distinguishing features of the Sanskrit language is an accurate specification of its grammar rules. The authoritative work on the Sanskrit grammar is by Pāṇini in the form of Aṣṭādhyāyī, meaning a collection of eight books (Pāṇini and Katre, 1987), which comprises a total of 3,959 *sūtras* (concise rules). Aṣṭādhyāyī gives an almost complete specification of the Sanskrit grammar. Due to the precise specification of these rules, the language has not undergone much modification. The rules of the language, though small in number, are precise and rich in their structure which allow the users of the language tremendous flexibility in their usage of the language. The verb-roots, which are less than 2000 in number, can be combined and modified according to the well defined rules to form new words, making a rich lexicon of size limited only by the creativity of the writer.

The precise and complete specification of the Sanskrit grammar also opens up the possibility of the development of computational tools to assist the students of the language or general readers to help translate and interpret Sanskrit texts. This has spawned a new and active interdisciplinary area of research called *Sanskrit Computational Linguistics*, which aims to use computational tools for automating the analysis of Sanskrit texts (Huet, 2003; Huet, 2009; Huet, 2005; Kumar et al., 2010; Bharati and Kulkarni, 2007; Goyal et al., 2009; Kapp and Malten, 1997; Goyal and Huet, 2013; Kulkarni, 2017; Jha, 2017b; ILTP, 2012; GM, 2017; Omkarananda, 2003; UBC, 2017). Due to the availability of a complete formal specification of the Sanskrit grammar, the development of a perfect Sanskrit parser seems to be the guiding factor behind this research. This necessitates the development of sub-tools like dictionaries, morphological analyzers, Sandhi splitters, and de-compounders without which successful Sanskrit parsing cannot be done.

The Cologne Sanskrit Dictionary Project (Kapp and Malten, 1997) aims to digitize the major bilingual Sanskrit dictionaries and provide easy access to the meanings of all the Sanskrit words which may be used by computer programs that help analyze Sanskrit texts. The most prominent among the remaining tools are (a) the Sanskrit Reader Companion (Goyal and Huet, 2013) by Inria which has tools for declension, conjugation, Sandhi splitting and merging along with word stemming (b) the Saṃsādhanī - A Sanskrit Computational Toolkit by University of Hyderabad (Kulkarni, 2017), which has tools for morphological analysis and generation, Sandhi splitting and merging, declension, conjugation and other form of word modifications and (c) the Sanskrit language processing tools developed at the Jawaharlal Nehru University (JNU) (Jha, 2017b), which comprise tools for Sandhi splitting and merging, declension, conjugation, POS tagger and other forms of word modifications.

Although researchers have been working in the field of Sanskrit Computational Linguistics for many years, there has been no effort to standardize, validate or critically evaluate the outcome of the work done so far. In this paper, we take a small step in the direction of standardizing and validating the research in this field. We examine the process by which two or more words combine to form a new word, a process known as *Sandhi*. The process of Sandhi is fundamental to the Sanskrit language as it enables the formation of new and more complex words using simpler words. Any computational tool for processing Sanskrit needs to be able to merge and split the words according to the rules of Sandhi. The correctness of any such tool critically depends on the correctness of its Sandhi processing.

We create a benchmark corpus called *SandhiKosh* that may be used by researchers to evaluate and validate the correctness and accuracy of their Sanskrit Sandhi and Sandhi-splitting tools. This corpus consists of examples of words along with their correct Sandhi-splitted root words. These examples have been categorized into the following five sub-corpora: (a) a list of 282 examples based on the Aṣṭādhyāyī rules; (b) a list of 150 examples hand picked from eleven well-known Sanskrit texts; (c) a list of 1432 examples taken from the most famous Sanskrit text Bhagavad-gītā; (d) a list of 10107 examples taken from digitized Sanskrit text at University of Hyderabad (Kulkarni, 2017); and (e) a list of 2700 examples taken directly from the Aṣṭādhyāyī text, which itself has been written in Sanskrit. Some of the examples were hand picked and manually verified for correctness while the other examples were created using the existing computational tools and validated computationally using a variety of methods.

We evaluated the three major Sanskrit Sandhi tools ((Jha, 2017a), (Kulkarni, 2017), and (Goyal and Huet, 2013)) using our SandhiKosh benchmark. Our results indicate the all these tools can benefit substantially from SandhiKosh. The SandhiKosh benchmark corpus will be freely available to researchers working in this area and we hope that it will lead to significant improvement in the state-of-the-art in the field of Sanskrit Computational Linguistics.

In Section 2. we describe the process of Sandhi in a little more detail and in Section 3. we describe the three major Sanskrit Computational Linguistics tools. The methodology followed during the creation of SandhiKosh has been described in Section 4. and the evaluation results are presented in Section 5.. We discuss possible future improvements in the Sanskrit Sandhi tools and in the SandhiKosh benchmark in Section 6. and conclude in Section 7..

2. Introduction to Sandhi in Sanskrit

Word formation in Sanskrit is centered around a root verb, modified by a suffix (and additionally a prefix in certain cases). Each of these three (roots, prefixes, and suffixes) represents a morpheme category, as these are the meaningful morphological units of the language and none of them can be further divided. Further, Sanskrit texts contain numerous words which are formed by the combination of two or more words. This process, generally known as *Sandhi*, takes place according to certain rules codified by the grammarian Pāṇini in his Aṣṭādhyāyī. The reverse process of getting back the component morphemes/words from the Sandhied words is known as *Sandhi Viccheda* or Sandhi splitting.

Interestingly, each of the two words *Sandhi* and *Viccheda* is itself made up of two components – *Sam* + *dhi* and *vi* + *cheda* respectively. *Sam* (meaning together) and *dhi* (meaning placement/location) combine to give *Sandhi* which means 'placed together/joined/merged'. *Vi* (meaning special) and *cheda* (meaning split/ breaking down) combine to give *Viccheda* which means special splitting (as opposed to simply splitting a word into each of its component letters). The Sandhi process is akin to that in some other languages, such as in English, *"come"* + *"-ing"* → *"coming"*, where we lose the additional "*e*" in the word *"come"* while merging. Another category of examples includes words such as "indirect", "impossible", and "irrelevant", where all these words have the same prefix as "in-", however, that got modified when merging with the root word. However, there is a very important difference between Sandhi in Sanskrit and such a combination process in English, as explained later in this section.

2.1. Conditions for Sandhi

Interestingly, the word *Sandhi* does not appear in any of the Aṣṭādhyāyī *sūtras* (concise rules). There are certain *sūtras* that are governed by a condition known as *saṃhitā* which as defined in *sūtra* 109 of Chapter 4 of Book 1, means "closest proximity of letters". These *sūtras* talk about the changes that take place when two letters are in "closest proximity". *Sūtras* 73 to 157 of Chapter 1 of Book 6 and all *sūtras* of Chapters 3 and 4 of Book 8 of Aṣṭādhyāyī are governed by the condition of *saṃhitā*. These rules are hereafter referred to as Sandhi rules. Thus, Sandhi is an umbrella term that is used to refer to sound changes that take place when two sounds are close enough.

The sound changes can take place in a variety of ways, depending not always only the two sounds (represented by the last character of the first word and the first character of the second word) combining but also sometimes on other factors, as described by Pāṇini in Aṣṭādhyāyī. The two sounds may merge to give a single sound, one of the two sounds (the former or the latter) may get changed/reduplicated before combining with the other, or even get elided. A new sound may also come in between.

2.2. Types of Sandhi

Sandhi can take place either within a word (internal Sandhi) or between two or more words (external Sandhi). Also, depending on whether the two letters that are being combined are vowels, consonants or the first of them is a visarga[1], the

[1] According to the Merriam-Webster dictionary Visarga refers to a "Sanskrit postvocalic sound or group of sounds produced by keeping the vocal organs above the glottis in the same position as

Criteria	Type	Explanation	Example	Analogy with English
Position	Internal	Root + Pref/Suffix	*bho + anam → bhavanam*	*come + ing → coming*
	External	Words combine	*tau + ekadā → tāvekadā*	*modify + ability → modifiability*
Type of character	Vowel	Vowel + Vowel	*hima + ālayaḥ → himālayaḥ*	*forgive + able → forgivable*
	Visarga	Visarga first	*punaḥ + janma → punarjanma*	No visarga in English
	Consonant	Other cases	*vṛkṣa + chāyā → vṛkṣacchāyā*	*forget + able → forgettable*

Table 1: Different types of Sandhi classification

Sandhi is classified as vowel, consonant or visarga Sandhi. The classification and the examples thereof have been summarized in Table 1. Similar examples from English language, wherever applicable, have also been provided.

2.3. Importance of Sandhi Splitting

Sandhi is very frequently encountered in classical texts of Sanskrit and these texts cannot be understood as long as the complex words (particularly the ones involving external Sandhi) are not broken down. There is an important difference between combination of words in English and that in Sanskrit. In English, combination of words is restricted by meaning and parts of speech involved. Thus, for example, in the sentence 'The regrettable decision of the chairman is now causing great harm to him', each of the words 'regrettable' , 'chairman' and 'causing' represents a combination, but combinations like 'Theregrettable' or 'regrettabledecision' or 'isnow' or worse 'Theregrettabledecisionofthechairmanisnowcausinggreatharmtohim' are simply not allowed. On the other hand, in Sanskrit, these are not only allowed but encountered very frequently. So while no combination is possible between the words of the sentence 'Ravi arrived in forest', all the words in the Sanskrit equivalent '*Raviḥ vane āgataḥ* ' can combine to form '*Ravirvanayāgataḥ*' (please note the changes at the boundaries of merging). Thus, Sandhi splitting is not only helpful but indispensable in the analysis of classical Sanskrit texts.

3. Existing Sandhi Tools

Over recent years a considerable amount of research has been carried out in the field of Sanskrit Computational Linguistics. A number of tools have been developed in this domain. As mentioned earlier, the development of a perfect Sanskrit parser seems to be the guiding factor behind this research, but this by itself necessitates the development of sub-tools like morphological analyzers, Sandhi splitters, de-compounders among others, without which successful parsing cannot be done.

In this section, we present the three most popular publicly available set of Sandhi and Sandhi Splitting tools. Given two words, the Sandhi process occurs as per the rules mentioned in the relevant sections of Aṣṭādhyāyī but there are multiple approaches to Sandhi splitting to get back the original words and we discuss the techniques used by these tools for the same. Comparison of these tools using the benchmark data set is provided in Section 5..

3.1. JNU Tools

The JNU Sandhi tool, known as the *Sanskrit Sandhi Generator*, was developed under the supervision of Prof. Girish Nath Jha. The corresponding Sandhi splitting tool, known as *Sanskrit Sandhi Recognizer and Analyzer* (Jha, 2017a) is specifically designed for vowel based Sandhi (Sachin, 2007). Using a dictionary of possible morphemes, this tool, at every location recursively checks for binary splits. To be a valid split, both the left and right split segments must be available in the dictionary. If the second segment has more than one sound marked for Sandhi, then only the first segment is matched against the dictionary.

3.2. UoH Sandhi Tools

These tools were developed at the Department of Sanskrit Studies, University of Hyderabad (UoH) under the guidance of Prof. Amba Kulkarni (Kulkarni, 2017). The Sandhi Splitting tool in this case also recursively breaks a word at every possible position and applies appropriate Sandhi rules to generate possible morpheme candidates and passes them through a morphological analyzer. The split words are considered as valid only if all its constituents are recognized by the morphological analyzer. Weights are assigned to the accepted candidates and then ranked based on the descending order of weights.

3.3. INRIA Tools

The Sandhi tool, known as *The Sandhi Engine*, was developed under the guidance of Prof. Gerard Huet at INRIA, France (Goyal and Huet, 2013). Of the three sandhi tools discussed here, this is the only tool which makes an explicit distinction between internal and external sandhi, giving the user both options to choose from. The external one corresponds to doing external sandhi in a deterministic fashion, with the most frequent rule, not taking into account optional rules. This is different from the UoH sandhi tool, that returns all possible solutions.The internal one is a rather adhoc processing, also deterministic, but corresponding more or less to INRIA's case generation with retroflexion. The other tool, called *The Sanskrit Reader Companion*, is actually more than a Sandhi splitting tool. It is designed to help a novice Sanskrit reader parse complex Sanskrit sentences. Sandhi splitting is only one part of the analysis. Initially, the word is analyzed to gather stems and their morphological parameters, such as permitted genders of nominal stems, allowed classes, and attested pre-verbs for roots. In the next stage, another round of stem generation is performed considering the various tenses, moods, absolutives, and participles in 10 varieties. Finally, inflexional morphology paradigms derive the inflected forms according to the

for the preceding vowel and continuing to expel air from the lungs but not vibrating the vocal cords."

morphological parameters, some of which are read from the word itself, while the others are defined in specific tables.

4. Creation of SandhiKosh

The SandhiKosh comprises of five sub-corpora that provide for a complete coverage of all the Sandhi rules of Aṣṭādhyāyī while at the same time are designed to give a good estimate of the accuracy of the Sandhi tools when applied to real Sanskrit texts.

4.1. Rule-based Corpus

The rule-based corpus is designed for checking for accuracy and completeness of the existing Sanskrit Sandhi tools. For this, all the rules of Aṣṭādhyāyī related to Sandhi were identified and unique examples corresponding to each of the rules were added to this corpus. Since some of the tools implement only a particular type of Sandhi, two separate datasets were created - one for internal Sandhi rules and the other for external Sandhi rules. At least one example for each rule in each of the two datasets was provided. If a rule applies to both internal and external Sandhi, an example was included in both corpora. This resulted in 150 examples for internal Sandhi and 132 examples for external Sandhi, with a total of 282 examples.

4.2. Literature Corpus

Some of the Sandhi rules are very common in Sanskrit texts while some other rules are rare. Therefore, although the rule-based corpus can give a good estimate of the completeness and validity of the Sandhi tools, it cannot estimate the accuracy of the tools when applied to real-world Sanskrit texts. In order to estimate the accuracy of Sandhi tools on classical and contemporary Sanskrit texts, 150 examples from a total of 11 different literary texts were handpicked to constitute the literature sub-corpus.

4.3. *Bhagavad-gītā* Corpus

In order to estimate the performance of Sandhi tools on some of the old Sanskrit classical texts, a corpus based on the *Bhagavad-gītā* was created. The *Bhagavad-gītā* is the best known and the most widely read and translated book from Sanskrit literature (Davis, 2014). It is organized into eighteen chapters comprising 700 verses. All the verses of its first nine chapters were critically analyzed and all the cases involving external Sandhi were split manually into their constituents leading to creation of a sub-corpus with 1430 examples.

4.4. UoH Corpus

The corpora described so far were manually created and therefore are comparatively small in size. These corpora may not estimate the performance of Sandhi tools accurately due to their small sizes and the particular examples and texts they are based on. A Sanskrit text with a different literary style may have very different statistical properties and give different performance when these Sandhi tools are applied. It was therefore very important to include a large corpus in the benchmark.

The University of Hyderabad (UoH) has digitized a large number of Sanskrit texts and made it freely available for researchers and general users (Kulkarni, 2017). For 39 of these texts, all the complex words are split according to the Aṣṭādhyāyī rules into smaller constituents and made available. This gives $113,913$ Sandhi splitting examples. While trying to make use of this corpus, we discovered several errors. These errors were filtered using a combination of the Cologne dictionaries (Kapp and Malten, 1997) (details omitted) leading to a list containing $9,368$ examples.

4.5. *Aṣṭādhyāyī* Corpus

Aṣṭādhyāyī is also written in Sanskrit and its split of words is also available at (SD, 2017). This was found to be another good source of Sandhi examples. However, even this source suffered from the limitation of insufficient splits. Moreover, a very significant number of splits were not located in any dictionary because of the way this text has been composed. Since the fundamental challenge is the insufficiency of splits, the splits which can undergo further splitting themselves are likely to be of greater length than fundamental morphemes. Thus using the length of the split words as a heuristic, a total of $3,959$ examples are reduced to $2,700$ where further splitting is applicable. Also, the results were noted for different values of the word lengths – 10, 20, 30, 40 and 50 (results omitted).

5. Evaluation Results

The results of evaluation of the Sandhi and the Sandhi Splitting tools on the corpora described above are presented in this section. These results hold true as of February 22, 2018 and may change in the future as and when the tools get updated.

5.1. Evaluation Methodology

We used a python-based automated evaluation method to automatically send web-requests to each of the tools, parse the HTML output and extract the relevant information automatically. The *requests* module of python was used to fetch the web page from source tool. The *BeautifulSoup* package was used to parse each of the web pages. Some of the tools generate multiple outputs for a single input along with a *filtered* output that is most likely to be correct. For all the three tools, we considered all the results instead of just the filtered output and marked the output to be correct if any of the results matched with the expected output of SandhiKosh. In case of Sandhi, several examples in the SandhiKosh corpus contain more than two words to be joined. However, all the three tools have provision for joining only two words at a time. For the examples where there were more than two words to be joined, we iteratively obtained the results from these tools to form the final Sandhi word.

5.2. Accuracy of Automated Evaluation

The automated evaluation methodology evaluated the accuracy of the tools by automatically extracting the results from the HTML output of the web-pages and then applying an exact string match to assessing the correctness of the results. This process of automatic evaluation may lead to reporting slightly higher error rate as compared to a manual evaulation process, due to differences in punctuation, spacing or other minor errors in the Sandhi process such

Corpus	Words	JNU	UoH	INRIA
Rule based - Internal	150	21 (14.0%)	36 (24.0%)	79 (52.7%)
Rule based - External	132	38 (28.8%)	57 (29.5%)	67 (50.8%)
Literature	150	53 (35.3%)	130 (86.7%)	128 (85.3%)
Bhagavad-gītā	1430	338 (23.64%)	1045 (73.1%)	1184 (82.1%)
UoH	9368	3506 (37.4 %)	7480 (79.8%)	7655 (81.7%)
Āstaadhyaayi	2700	455 (16.9%)	1752 (64.9%)	1762 (65.2%)

Table 2: Sandhi accuracy obtained by the three different Sandhi Tools available in the literature.

as omission of *visarga* etc. To assess this more systematically, we took a small corpus and evaluated it manually on all of the three tools and then compared the results to automated evaluation. It was found that for the UoH and JNI tools, the difference between automated and manual evaluation was small with manual evaluation reporting only marginally higher accuracy than the automated evaluation. However, in case of the INRIA tool, the difference was found to be slightly more but always less than 23%.

5.3. Sandhi Tools

The performance of Sandhi tools on SandhiKosh is presented in Table 2. The INRIA tool provides an option to select internal or external Sandhi that needs to be applied while creating the joined word. Since the SandhiKosh does not have information about internal or external Sandhi (except for the rule based corpus), we evaluated the entire corpus with both the options on the INRIA tool. If any of the option gives the correct word merging, the Sandhi is marked as correct. Thus, the results are presented by combining the results of both internal and external Sandhi options in the most optimistic manner.

For the rule-based (internal & external) corpus INRIA is the best performing tool at 51.8% accuracy followed by UoH and JNU Sandhi tools. On the Literature corpus, the accuracy of INRIA and UoH tool is comparable whereas that of JNU is substantially lower at 53%. These trends are consistent for rest of the corpora as well with INRIA and UoH tool performing at similar levels of accuracy (ranging from 23 to 65%).

5.4. Sandhi Splitting Tools

For Sandhi splitting, the INRIA tool as well as the UoH tool gives multiple possible splits of a given word. For the evaluation purposes, we examine all the possibilities given by these tools and mark the Sandhi splitting as correct if the correct split has been given as one of the options. In this way, we combine the multiple options given by these tools in the most optimistic manner.

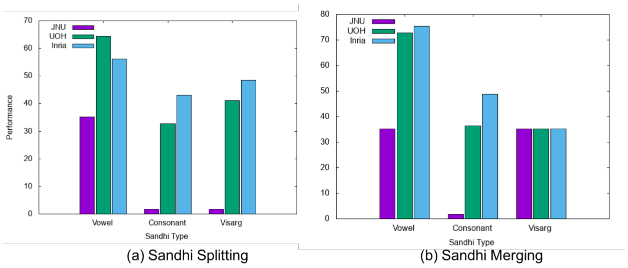
The accuracy of Sandhi splitting tools on SandhiKosh is presented in Table 3. On the rule-based (internal & external) corpus, the UoH tool performs at 26.6% accuracy whereas the INRIA and JNU tools perform respectively at 14.5% and 11.4%. The JNU tool performs the worst for all the corpora, whereas INRIA tool performs best on Bhagavad-gītā and Aṣṭādhyāyī corpora and UoH tool performs best on the Literature and UoH corpora.

Corpus	Words	JNU	UoH	INRIA
Rule based - Internal	150	10 (6.8%)	27 (18.0%)	3 (2.0%)
Rule based - External	132	22 (16.9%)	48 (36.4%)	38 (29.2%)
Literature	150	13 (8.7%)	98 (65.3%)	101 (67.3%)
Bhagavad-gītā	1430	67 (4.9%)	650 (45.5%)	962 (70.8%)
UoH	9368	934 (10.0%)	6393 (68.2%)	6490 (69.3%)
Āstaadhyaayi	2700	18 (0.7%)	263 (9.7%)	510 (18.9%)

Table 3: Sandhi splitting accuracy obtained by the three different splitting tools available in the literature.

5.5. Sandhi type based performance

The Sandhi splitting and merging performance is evaluated on three different types of Sandhi – vowels, consonants, and visarga Sandhi types and the results are shown in Figure 1 (for all the corpora put together).



(a) Sandhi Splitting

(b) Sandhi Merging

Figure 1: Performance of different splitting tools on the combined corpora for three Sandhi types.

The JNU tools which presently support Vowel Sandhi only are barely able to give any correct results on Consonant and Visarga Sandhi types as expected. However, on the Vowel Sandhi type, the accuracy of the JNU tools is less than that of the UoH and INRIA tools for Sandhi splitting but comparable in the case of Sandhi merging.

6. Discussion

We draw attention to some of the reasons as to why the Sandhi splitting tools are not able to get the correct splits.

- **Rules not implemented:** Some of the rules have not been implemented by one or more of the three splitters. For example, none of the three splitters is able to do the following split: *sa yogī* → *saḥ* + *yogī*.

  The *sūtra* which applies in this case is 6.1.132 (*sūtra* 132 in first chapter of book 6 of Aṣṭādhyāyī) as per which the visarga *ḥ* of the words *saḥ* and *eṣaḥ* is elided when they combine with a word beginning with a consonant. Thus, sandhi-splitting in this case involves restoring the visarga in both the cases when the forms *sa* and *eṣa* are encountered before a word beginning with a consonant. As mentioned earlier, none of the splitters has implemented this string-match condition.

There are other *sūtras* similar to this, in which the sandhi phenomenon is not restricted to a focus only on two letters but also extends to the entire word/string under consideration.

- **Optional rules:** There are some rules which are optional in nature and less frequently used. The following case is an example where none of the three splitters is able to detect the correct split: *kumāri atra* → *kumārī* + *atra*. This is because *kumārī* + *atra* generally leads to *kumāryatra* through the *sūtra* 6.1.77 but the application of another *sūtra* 6.1.127 (which is less frequently applied) leads to the optional result *kumāri atra*. Getting back the correct split would require the reverse application of this rule.

- **Cascading effect:** There are some rules in which the result of combination of two letters may create the possibility of another sandhi, when a suitable context exists. A change may also occur far beyond the merging boundaries of the two words. For example, in *ṣaṭ* + *navatiḥ* → *ṣaṇṇavatiḥ*, the *n* of *navatiḥ* changes to *ṇ*, which then causes the change of *ṭ* of *ṣaṭ* to *ṇ* , thus resulting into *ṣaṇṇavatiḥ*. Thus, to get back the original words, both of these changes will have to be undone.

- **Validation problem:** The process of Sandhi splitting involves splitting at different potential locations, and validating the splits to check which one of them is correct. If the set used for validation is not complete, even correct splits may sometimes not be validated. For example, in *a* + *chedyaḥ* → *acchedyaḥ*, none of the three splitters performed correctly in the beginning because *a* may not have been validated as a proper split. However, the INRIA Sanskrit Reader Companion was updated recently to take care of this condition, when this was brought to the notice of the tool developers.

- **Compounding effect:** The process of compounding, due to which words come together without there necessarily being a change when they merge, also creates problems. While the UoH and the INRIA tools do have the provision of decompounding along with Sandhi splitting, the JNU splitter does not have a way to do both together. For example, in *lakṣyasyārthatvavyavahārānurodhena* → *lakṣyasya* + *arthatvavyavahāra* + *anurodhena* the second split is not validated without decompounding, and thus even though, only vowel Sandhis are involved, the JNU splitter is not able to correctly split the word.

It is to be noted that our benchmarking corpus SandhiKosh is expected to undergo refinements in the future as there is scope for its improvement. In subsequent releases of the corpus, we aim to deal with the optional rule scenarios by presenting all valid splits and/or merged words for a given corpus entry. We also plan to provide a trace of the Aṣṭādhyāyī rules applied in the process of Sandhi for each example. Along with these, we also plan to propose standardizations for word spacing, the usage of avagraha and the presence of anuswara to make the evaluation process more rigorous and extensive.

7. Conclusion

Standardization of benchmarks has a profound impact on the corresponding field. The benchmarks shape a field by giving an objective yardstick against which different researchers strive to improve their performance, thereby leading to a faster overall development. Benchmarks such as top 500 supercomputers (Dongarra et al., 1997) and SPEC (Henning, 2006) in the area of computing have shaped how computers were designed and built.

In this paper we have attempted to design the first benchmark called SandhiKosh in the area of Sanskrit Computational Linguistics. SandhiKosh provides a way to measure the accuracy and completeness of Sanskrit Sandhi tools. The examples in SandhiKosh were selected usign five different methods in order to provide a corpus that is complete as well as able to reflect the performance of Sandhi tools on actual Sanskrit literature.

We evaluated the performance on three available Sanskrit Sandhi tools on SandhiKosh. Our results indicate that the best performing Sandhi merging tools give an accuracy of in the range of 50-60% where as the worst performing tools result in 20-30% accuracy. For Sandhi splitting, which is a harder problem, the best tools give an accuracy of 50-60% where as the worst performing tools give an accuracy between 5-15%.

SandhiKosh will be freely available to researchers and we hope that it will lead to faster overall progress in the area of Sanskrit Computational Linguistics.

Acknowledgements

We thank Prof. Amba Kulkarni (University of Hyderabad), Prof. Gerard Huet (INRIA, France) and Prof. Girish Nath Jha (Jawaharlal Nehru University, Delhi), for providing insightful comments and useful improvements to our benchmark corpus. We would also like to thank Samskrita Bharati and its teachers such as Pt. Narayan Dutt Suvedi and Dr. Sachin Kathale who, without any financial or any other expectations, taught Sanskrit to some of the authors. They patiently kindled the interest in the language by demonstrating its Maths-like precision along with its poetic beauty which inspired us to explore and work on this further. This work would not have been possible without the seeds of love and selfless service sown by these teachers.

References

Bharati, A. and Kulkarni, A. (2007). Sanskrit and computational linguistics. In *First International Sanskrit Computational Symposium. Department of Sanskrit Studies, University of Hyderabad*.

Davis, R. H. (2014). *The Bhagavad Gita: A Biography*. Princeton University Press.

Dongarra, J. J., Meuer, H. W., Strohmaier, E., et al. (1997). TOP500 supercomputer sites. *Supercomputer*, 13:89–111.

Ghosh, A. (2015). Sanskrit fever grips Germany. Daily Mail Online `http://www.dailymail.co.uk/`, Published 14^{th} April, 2015.

GM. (2017). Sanskrit online tools. Green Message, `http://greenmesg.org/sanskrit_online_tools/`.

Goyal, P. and Huet, G. (2013). Completeness analysis of a sanskrit reader. In *Proceedings, 5th International Symposium on Sanskrit Computational Linguistics. DK Printworld (P) Ltd*, pages 130–171.

Goyal, P., Arora, V., and Behera, L. (2009). Analysis of Sanskrit text: Parsing and semantic relations. In *Sanskrit Computational Linguistics*, pages 200–218. Springer.

Henning, J. L. (2006). SPEC CPU2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News*, 34(4):1–17.

HT. (2007). Popularity of Sanskrit on rise in US, Europe. Hindustan Times News, Published 10^{th} July, 2007, `https://www.hindustantimes.com/`.

Huet, G. (2003). Towards computational processing of Sanskrit. In *International Conference on Natural Language Processing (ICON)*.

Huet, G. (2005). A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *Journal of Functional Programming*, 15(4):573–614.

Huet, G. (2009). Formal structure of Sanskrit text: Requirements analysis for a mechanical Sanskrit processor. In *Sanskrit Computational Linguistics*, pages 162–199. Springer.

ILTP. (2012). Indian language technology proliferation and deployment center, sandhi-splitter. `http://tdil-dc.in/san/sandhi_splitter/index_dit.html`.

Jha, G. N. (2017a). Sanskrit sandhi recognizer and analyzer. `http://sanskrit.jnu.ac.in/sandhi/viccheda.jsp`.

Jha, G. N. (2017b). Special centre for Sanskrit studies. `http://sanskrit.jnu.ac.in/index.jsp`.

Kapp, D. B. and Malten, T. (1997). Report on the Cologne Sanskrit Dictionary Project. In *10th International Sanskrit Conference, Bangalore*.

Kulkarni, A. (2017). Saṃsādhanī - A Sanskrit computational toolkit. `http://sanskrit.uohyd.ac.in/scl/`.

Kumar, A., Mittal, V., and Kulkarni, A., (2010). *Sanskrit Compound Processor*, pages 57–69. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mitra, V. L. (1891). Yoga Vashistha Maharamayana of Valmiki (4 vols). *Calcutta: Bannerji*.

Omkarananda, A. H. (2003). Itranslator. `https://www.oah.in/Sanskrit/itranslator2003.htm`.

Pāṇini and Katre, S. M. (1987). *Aṣṭādhyāyī of Pāṇini*. University of Texas Press.

Prasada, R. (1998). *Patanjali's yoga sutras*. South Asia Books.

Raub, J. A. (2002). Psychophysiologic effects of hatha yoga on musculoskeletal and cardiopulmonary function: a literature review. *The Journal of Alternative & Complementary Medicine*, 8(6):797–812.

Sachin, K. (2007). Sandhi splitter and analyzer for Sanskrit (with reference to ac Sandhi). *M. Phil. degree at SCSS, JNU (submitted, 2007)*.

Saraswati, N. (2013). *Gheranda Samhita*. Yoga Publications Trust, Munger, Bihar, India.

SD. (2017). Panini Ashtadhyayi sutras with commentaries: Sortable index. `http://sanskritdocuments.org/learning_tools/ashtadhyayi/`.

Singh, P. (1915). *Hatha yoga pradipika*. Dev Publishers.

SJJS. (2017). St James Junior School, Languages Curriculum. `https://www.stjamesschools.co.uk/juniorschools/curriculum/academic/languages/`.

Swarupananda, S. (2016). *Srimad Bhagavad Gita*. Advaita Ashrama (A publication branch of Ramakrishna Math, Belur Math).

(2017). UBC Sanskrit learning tools. `https://ubcsanskrit.ca/`.

Uebelacker, L. A., Epstein-Lubow, G., Gaudiano, B. A., Tremont, G., Battle, C. L., and Miller, I. W. (2010). Hatha yoga for depression: Critical review of the evidence for efficacy, plausible mechanisms of action, and directions for future research. *Journal of Psychiatric Practice®*, 16(1):22–33.

UN. (2014). United Nations Resolution 69/131. *General Assembly of the United Nations*. Available from `http://www.un.org/en/events/yogaday/`.

Yang, K. (2007). A review of yoga programs for four leading risk factors of chronic diseases. *Evidence-Based Complementary and Alternative Medicine*, 4(4):487–491.