N-gram-Based Preprocessing for Sandhi Reversion in Vedic Sanskrit

Yuzuki Tsukagoshi and Ikki Ohmukai
The University of Tokyo / Tokyo, Japan
{yuzuki, i2k}@l.u-tokyo.ac.jp

Abstract

This study aims to address the challenges posed by sandhi in Vedic Sanskrit, a phenomenon that complicates the computational analysis of Sanskrit texts. By focusing on sandhi reversion, the research seeks to improve the accuracy of processing Vedic Sanskrit, an older layer of the language. Sandhi, a phonological phenomenon, poses challenges for text processing in Sanskrit due to the fusion of word boundaries or the sound change around word boundaries. In this research, we developed a transformer-based model with a novel n-gram preprocessing strategy to improve the accuracy of sandhi reversion for Vedic. We created character-based n-gram texts of varying lengths (n = 2, 3, 4, 5, 6) from the Rigveda, the oldest Vedic text, and trained models on these texts to perform machine translation from post-sandhi to pre-sandhi forms. In the results, we found that the model trained with 5-gram text achieved the highest accuracy. This success is likely due to the 5-gram's ability to capture the maximum phonemic context in which Vedic sandhi occurs, making it more effective for the task. These findings suggest that by leveraging the inherent characteristics of phonological changes in language, even simple preprocessing methods like n-gram segmentation can significantly improve the accuracy of complex linguistic tasks.

1 Introduction

1.1 Sandhi in Sanskrit

Vedic Sanskrit represents an earlier stage of the Sanskrit language, characterized notably by its unique system of accentuation, where the meaning of words in Vedic can vary depending on the position of the accent.

Sandhi refers to a phonological/phonetic phenomenon wherein certain conditions induce changes in the sounds of adjacent words or morphemes (Macdonell, 1910). This phenomenon is divided into two types: external sandhi, which involves sound changes at the junctions between words in a sequence, and internal sandhi, which affects the internal structure of words, such as in the derivation of nouns and verbs.

For instance, when the words *yás* ("who/which", a relative pronoun in the nominative singular masculine), *hatvā́* ("having killed", absolutive of the verb *han*), and *áhim* ("a snake", a noun in the accusative singular) consecutively in a sentence, they have a surface form like *yó hatvā́him* (Rigveda 2.12.3a) meaning "[Indra,] who [...] after killing the snake". The final two sounds of *yás* change to -ó before the voiced sound h-. Additionally, the final vowel -ā́ of *hatvā́* and the initial vowel *á-* of *áhim* merge into -ā́-. Conversely, when the same word *hatvā́* is followed by *pr̥thivyā́m* ("on the earth"), it becomes *hatvā́ pr̥thivyā́m* (Rigveda 1.100.18b), meaning "After [he] struck [Dasyus and Śimyus, felled them] on the earth." In this case, the final vowel -ā́ of *hatvā́* does not alter before the voiceless stop p-. Hence, external sandhi varies according to the boundary sounds of adjacent words in a sentence. Internal sandhi, in contrast, can alter the sound at the morpheme boundary, as illustrated by the example of the genitive singular form of the stem *tanú-* ("body") is *tanv-às*. Internal sandhi also undergoes variations depending on the adjacent sounds, similar to external sandhi.

In general, Sanskrit texts preserve the forms resulting from sandhi rules as they are. Therefore, to analyze Sanskrit texts, it is essential to identify the original word forms prior to the sound changes imposed by sandhi.

Despite the precise definition of sandhi rules, reversing the transformations to recover pre-sandhi forms poses significant challenges, whether done manually or via computational methods. To accurately revert *hatvā́him* back to the two separate words *hatvā́* and *áhim*, one must have knowledge of the word forms: the absolutive *hatvā́* of the verb *han* ("kill") and the accusative singular *áhim* of the noun *áhi-* ("snake"). When attempting to segment the phonetic sequence h-a-t-v-ā́-h-i-m into two parts, one must first determine the necessity of splitting. If splitting is required, a sequence containing a long vowel *-ā-* without consideration of accent, may yield four potential outcomes: *-ă ă-*, *-ă ā-*, *-ā ă-*, and *-ā ā-* (For clarity, short vowels are indicated with ă.). Furthermore, when taking accent into account, numerous additional possibilities arise, including sequences where both parts bear an accent, or sequences where only one part is accented, e.g. *ihā́sti* = *ihá asti*, *índrā́* = *índra ā́*.

1.2 Necessity of Sandhi Reversion

The process of restoring word sequences in a text from their post-sandhi forms back to their original, pre-sandhi forms is termed *sandhi reversion* or the act of *reverting sandhi* in this paper. Sandhi reversion is essential for accurately understanding and analyzing Sanskrit texts, as the phenomenon of sandhi causes phonetic changes that can obscure the original word boundaries. To perform sandhi reversion effectively, it is necessary to consider not only the given phonetic sequence but also the morphological and syntactic context in which these forms appear.

As discussed in the previous section (section 1.1), Sanskrit texts are predominantly written in their post-sandhi forms. This convention presents a challenge for linguistic analysis and computational processing, as it complicates the direct extraction and identification of individual words from the text. Thus, sandhi reversion becomes a crucial preprocessing step in many aspects of Sanskrit language processing, enabling scholars and computational tools to work with the underlying word forms rather than their modified, surface-level representations.

The reversion of external sandhi, in particular, which involves converting a continuous sentence into a series of discrete words, is vital in various domains such as natural language processing (NLP), philology, and linguistics. External sandhi impacts word recognition and syntactic analysis, and its accurate reversion is necessary to ensure proper word segmentation and syntactic parsing in these fields. Therefore, this study focuses on external sandhi. For the remainder of this paper, the term "sandhi" will refer exclusively to external sandhi, reflecting its primary importance for our objectives.

2 Related Work

Given the vast corpus of Sanskrit texts, manually reverting all instances of sandhi is impractical and time-consuming. As a result, numerous computational methods for sandhi reversion have been developed over the years to automate this process.

Recent research has shown that machine learning-based sandhi reversion is more accurate than previous rule-based approaches. The following sections review major studies that have employed machine learning techniques for sandhi reversion.

2.1 seq2seq2 + Attention / Classic

Reddy et al. (2018) approached sandhi reversion by treating a post-sandhi sentence as the source text and a pre-sandhi sentence as the target text, effectively framing the problem as a translation task. They applied a sequence-to-sequence (seq2seq) model with attention mechanisms, utilizing SentencePiece (Schuster and Nakajima, 2012) for preprocessing. This approach facilitated the creation of a translation model that could convert post-sandhi sentences back to their pre-sandhi forms. The study utilized a dataset (Krishna et al., 2017) of approximately 100,000 sentences to train the model.

The method involved segmenting sentences as demonstrated below, where spaces (represented by underscores, _) are treated as individual characters, and each resulting segment is considered a word within the restructured sentence:

(original) putraṁ vaṁśakaraṁ rāma nr̥pasaṁnidhau

(sentencepiece) _putraṁ _vaṁś akar aṁ_rāma nr̥pa [sic.] [1] saṁnidh au

2.2 RNN + CNN / Classic + Vedic

Hellwig and Nehrdich (2018) developed a neural network model that combines Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) using character-based n-grams to address the challenge of sandhi reversion in Sanskrit texts. Their study evaluated the performance of various models, including the seq2seq approach by Reddy et al. (2018), and demonstrated that their CNN+RNN model outperformed previous methods in terms of accuracy and effectiveness for sandhi reversion.

[1] An underscore should be placed before nr̥pa.

The dataset used in the study comprised both Classical Sanskrit and Vedic Sanskrit texts. However, it is important to note that the Vedic Sanskrit texts in the dataset had their accent marks removed. As a result, the sandhi reversion model developed in this study cannot be directly applied to Vedic Sanskrit texts that include accent marks. In Vedic Sanskrit, where the meaning can change based on the position of the accent within a word, the absence of accent information in the dataset prevents accurate reconstruction of the original accentuation patterns. Thus, restoring the accents in Vedic Sanskrit from texts without them is not feasible using this model.

3 Method

This study addresses the task of translating text from post-sandhi to pre-sandhi forms, building upon the approaches discussed in the related work section (section 2). The objective is to develop a model that accurately reverses the sandhi processes, thereby restoring the original word boundaries in Vedic Sanskrit texts.

3.1 N-gram translation

In this study, a new "sentence" is treated as one that is composed of character-based n-grams of a given sentence. A character-based n-gram divides the sentence into n-character units.

For example, *y/o+hatv/āhim* (= *yó hatvā́him*) shown in the section 1.1. When n = 3, for both post-sandhi *y/o+hatv/āhim* and pre-sandhi *y/aḥ hatv/ā /ahim* the texts result in the following n-grams:

> y/o /o+ o+h +ha hat atv tv/ v/ā /āh āhi him (post-sandhi)
>
> y/a /aḥ aḥ+ ḥ+h +ha hat atv tv/ v/ā /ā+ ā+/ +/a /ah ahi him (pre-sandhi)

In these examples, acute accent marks are represented by slashes, and spaces by plus signs.

This method differs from the n-gram method of Kitagawa and Komachi (2018) and its derivative by Hellwig and Nehrdich (2018). In their methods, each character-based input layer simultaneously embeds n-gram strings starting/ending with that character.Consequently, all n-grams within a specified range are used as input for a single model. In contrast, our method treats each segment arranged in n-grams as a word unit.

Character-based n-gram texts were generated for n = 2, 3, 4, 5, 6. In addition to these n-gram variations, an original text was prepared with no further processing beyond transliteration. These six types of texts were utilized as training data, as described in section 3.2. The model, detailed in Section 3.3, was then trained on each set of texts to perform machine translation from post-sandhi to pre-sandhi forms. This comprehensive approach allows for the assessment of how different n-gram lengths affect the model's ability to accurately perform sandhi reversion.

3.2 Training datasets

The primary text used in this study is the electronic version of the Rigveda provided by Martínez García and Gippert (1995). The Rigveda, a central text within the Vedic corpus, exists in several versions, including the Saṁhitāpāṭha, which represents the post-sandhi form, and the Padapāṭha, which provides a pre-sandhi form. The Padapāṭha is a later interpretative text based on the Saṁhitāpāṭha, created to disambiguate the text by resolving the effects of sandhi.

Due to the inherent ambiguities involved in sandhi reversion, the Padapāṭha is considered one of several possible interpretations of the R̥gveda Saṁhitā. Despite this, having paired post-sandhi and pre-sandhi texts is invaluable for supervised machine learning tasks. With approximately 10,000 verses, the Rigveda provides a substantial and well-suited dataset for training models to perform sandhi reversion.

3.3 Model

In this study, we address the problem of sandhi reversion as a translation task, converting post-sandhi text (Saṁhitāpāṭha) into pre-sandhi text (Padapāṭha). To accomplish this, we developed a transformer-based translation model.

Our approach focuses on preprocessing and the preprocessing strategy is designed to be independent of the specific machine translation architecture, making it compatible with the state-of-the-art architecture. [2]

4 Results

Table 1 presents the Precision, Recall, and F1 scores for models with six different types of text: one model trained on the original, unprocessed text (denoted as "Word") and five models trained on

[2]The model, dataset, and associated scripts is available at `https://github.com/Yuzki/SktTool`.

character-based n-gram texts with varying length (n = 2, 3, 4, 5, 6). These metrics provide a comprehensive evaluation of the model's performance in translating post-sandhi to pre-sandhi forms across different preprocessing strategies.

For comparative purposes, Table 2 displays the performance metrics reported by Hellwig and Nehrdich (2018) for their combined CNN and RNN model, alongside results from earlier studies.

	Precision	Recall	F1
Word	95.0	96.6	95.8
2-gram	87.8	88.3	88.1
3-gram	94.8	95.9	95.4
4-gram	95.8	97.2	96.5
5-gram	**96.0**	**97.7**	**96.8**
6-gram	95.3	97.2	96.2

Table 1: Precision, Recall, and F1 of each n-gram text and the original text.

	Precision	Recall	F1
Hellwig (2015b)	91.8	91.8	94.8
Reddy et al. (2018)	90.2	88.4	93.3
Transformer 5K	94.9	94.5	96.5
$\text{rcNN}^{\text{split}}_{\text{short}}$	94.6	94.8	96.7

Table 2: Excerpt from Table 3 in Hellwig and Nehrdich (2018).

The training and evaluation times for each model are not a primary focus of this study, because the computational efficiency is less of a concern for Sanskrit, an ancient language with a relatively fixed corpus, compared to the modern languages with increasing linguistic resources.

The results indicate that the sandhi reversion model trained on 5-gram text achieves the highest Precision, Recall, and F1 scores among all the models tested. Notably, these scores are slightly higher than those reported for the most accurate sandhi reversion model using the combined RNN + CNN approach by Hellwig and Nehrdich (2018), even though the datasets used in the two studies differ.

5 Limitations

While the proposed n-gram-based preprocessing method showed promising results, several limitations remain. First, the dataset used in this study was limited to the Rigveda, and the model's effectiveness on other Vedic or Classical Sanskrit texts remains to be evaluated. Additionally, although the 5-gram model performed well, this approach may not generalize to texts with different phonological structures, such as accentless Sanskrit texts. Another limitation is the reliance on the Padapāṭha as the pre-sandhi form, which is one of several possible interpretations of the Rigveda.

6 Conclusion

The results of this study demonstrate that some models utilizing n-gram preprocessing outperform the model with no special text processing. The highest accuracy is particularly achieved when n = 5 (Table 1).

This n value is thought to be related to the phonological environment in which sandhi occurs. The maximum number of phonemes in the environment where sandhi occurs is five: /a s + X -; where / denotes an accent, + denotes a space, and X represents an arbitrary voiced sound. Therefore, when n = 5, the n-gram text adequately includes this sequence, potentially resulting in the highest accuracy. However, it is predicted that n = 4 will yield the highest accuracy for texts in Classical Sanskrit, which does not have accents.

By demonstrating the effectiveness of n-gram preprocessing, this study shows that the approach is independent of the specific translation architecture used. This flexibility implies that our method can be adapted to future innovations in translation tasks, regardless of the underlying architecture.

Although the corpus of Vedic texts is smaller than that of Classical texts, Vedic language processing remains in its early stages. Implementing sandhi reversion as a preprocessing technique significantly advances Vedic NLP by facilitating tasks such as named entity recognition and intertextual comparison based on word occurrences.

Furthermore, this n-gram segmentation method is not restricted to the sandhi reversion problem in Vedic Sanskrit; it is also applicable to word segmentation challenges in Classical Sanskrit, which lacks accents, as well as in other languages. In these contexts, using n-grams with a size close to or matching the maximum string length of the relevant phonetic sequences in each language is expected to yield high accuracy.

References

Oliver Hellwig and Sebastian Nehrdich. 2018. Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2754–2763. Association for Computational Linguistics.

Yoshiaki Kitagawa and Mamoru Komachi. 2018. Long short-term memory for japanese word segmentation. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*. Association for Computational Linguistics.

Amrith Krishna, Pavan Kumar Satuluri, and Pawan Goyal. 2017. A dataset for sanskrit word segmentation. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 105–114. Association for Computational Linguistics.

Arthur Anthony Macdonell. 1910. *Vedic grammar*. Verlag von Karl J. Trübner.

F. J. Martínez García and J. Gippert. 1995. Plain text retrieval, thesaurus indogermanischer text- und sprachmaterialien. Online; accessed 17 October 2022.

Vikas Reddy, Amrith Krishna, Vishnu Sharma, Prateek Gupta, Vineeth M R, and Pawan Goyal. 2018. Building a word segmenter for sanskrit overnight. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.