文字単位 n-gram の翻訳によるヴェーダ・サンスクリットの連声解除

塚越 柚季 (東京大学大学院人文社会系研究科)

概要: 本研究では，サンスクリット語の古層言語であるヴェーダ語を対象に，深層学習による連声解除を行う．古典サンスクリット語と異なり，アクセントが弁別的であるヴェーダ語には，それに適した連声解除モデルが必要である．本研究において提案する手法は，前処理の 1 つとしてテキストを文字単位 n-gram への置き換えることである．適切な n-gram への置き換えが精度向上に寄与し，その結果は既存のモデルに勝ることを示す．また，n の値がヴェーダ語における連声の音環境に関係した値であることを示す．これにより，古典サンスクリット語をはじめとする言語の単語分割問題にも応用可能であることが示唆される．

キーワード: サンスクリット，ヴェーダ語，連声，サンディ，深層学習，テキスト処理

Vedic Sanskrit word segmentation using a method of character n-gram translation

Yuzuki Tsukagoshi (Graduate School of Humanities and Sociology, University of Tokyo)

Abstract: This study proposes a method to improve word segmentation for Vedic Sanskrit by deep learning. Vedic, different from Classical Sanskrit, has a distinctive accent system and requires a suitable segmentation model. The method proposed in this study is to replace the text with character n-grams as one of the preprocessing steps. We show that the appropriate n-gram replacement improves accuracy and outperforms existing models. It is also shown that the value of n is related to the phonological environment of Sandhi rules in Vedic. This suggests that the model can be applied to word segmentation problems in other languages, including Classical Sanskrit.

Keywords: Sanskrit, Vedic, Sandhi, deep learning, text processing

1. サンスクリット

サンスクリットとは，インド・ヨーロッパ語族インド・イラン語派インド・アーリア語群に属する古典語である．インド亜大陸およびその周辺で用いられた言語であり，主な資料は文献として残る．サンスクリットは，古典サンスクリットと，より古いヴェーダ・サンスクリット (ヴェーダ語) とに大きく分けられる．ヴェーダ語が持つ最も顕著な特徴は，アクセントが弁別的であることである．つまり，ヴェーダ語では単語のどの位置にアクセントがあるかによって意味が異なる．文献の読解をはじめとしてヴェーダ語を扱う際には，古典サンスクリットと異なり，アクセントに注意しなければいけない．

1.1 連声

連声とは，特定の条件下の音連続が変化する現象である [1]．文中の単語の連続において，語末および/あるいは語頭の音が変化する外連声と，名詞や動詞などの派生において生じる内連声との 2 種に分けられる．

例えば，*yás*「[関係代名詞 (主格単数男性)]」, *hatvá*「殺してから」, *áhim* 「蛇を」の 3 語がこの順番で文中に現れるとき *yó hatvā́him* (『リグ・ヴェーダ』2 巻 12 歌 3 節 a 行) 「蛇を殺してから [· · · するインドラ神が]」となる．*yás* の語末の 2 音は，有声音 h- の前で -ó と変化する．また，*hatvá* の語末の母音 -á と *áhim* の語頭の母音 á- が融合して-á- と変化する．一方，同じ *hatvá* という単語が *pr̥thivyā́m*「大地において」と連続すると，*hatvá pr̥thivyā́m* (『リグ・ヴェーダ』1 巻 100 歌 18 節 b 行)「[ダシユとシミユを] 討ち，[飛び道具によって] 大地に [倒した]」となる．このとき，

©2022 Information Processing Society of Japan

hatvā́ の語末の母音 -ā́ は，無声音 p- の前で変化しない．このように，外連声は，文中で隣接する単語の境界部の音により変化の仕方が異なる．一方，内連声は，語幹 *tanú-*「体」の属格単数形が *tanv-às* となるように，形態素の境界部において音変化しうる．例は示さないが，内連声も外連声と同様に隣接する音により異なる変化をする．

一般的にサンスクリット文献では，連声規則が適用された語形がそのまま残されている．そのため，正しく語を分析するためには，連声による音変化前の正しい語の形を把握する必要がある．

連声によって音変化する規則は厳密に定められている一方，音変化した語の連続から，音変化する前の語の連続に戻す作業は，人であれ機械であれ困難な場合がある．先の例の *hatvā́him* を *hatvā́* と *áhim* の 2 語にこの形で正しく戻すには，動詞 *han*「殺す」の絶対分詞 *hatvā́*，名詞 *áhi-* の単数対格形 *áhim* という語形を知っておく必要がある．h-a-t-v-ā́-h-i-m という単純な音連続を機械的に 2 つに分割する[*1]ときを考える．まずアクセントを考慮せず長母音 *-ā-* を含む連続を分割するとき，*-ă ă-*, *-ă ā-*, *-ā ă-*, *-ā ā-* の 4 通りが考えられる[*2]．さらにアクセントを考慮すると，*-ă̌ ă̌-* の両方にアクセントがあるもの (= この例)，いずれか一方にアクセントがあるもの[*3]と数多くの候補を想定できる．

1.2 連声解除の必要性

連声規則が適用された文中の語の連続を，連声規則が適用される前の語形に戻すことを，ここでは連声解除と言うことにする．連声解除のためには，与えられた音連続の他に形態や統語の情報を要する．前節 (1.1節) で述べたように，サンスクリット文献では，一般的に連声適用後の文が伝えられているため，サンスクリットを対象に言語処理を行う際，多くの場合に前処理として連声を解除することが必要である．特に外連声の解除によって，少なくとも文を独立した単語の列に変換することは，様々な場面において要求される．本研究では，連声の中でも外連声に着目する．よって，これ以降における連声は，外連声に限定されたものである．

2. 連声解除についての先行研究

膨大な数のサンスクリット文献を分析する際に，全てを人の手で連声を解除するのは現実的ではない．そのため，今までにコンピュータによる連声解除の手法が数多く提案されてきた．とりわけ，近年幅広い分野で活躍する機械学習を用いた連声解除の研究が盛んに行われている．ルールベースで連声解除を行ってきた既存の研究よりも，機械学習を用いた連声解除は精度が良い．以下の節に，機械学習による連声解除の主要な研究を挙げる．

2.1 seq2seq2 + Attenstion / Classic

Reddy et al.[2] は，連声後の文を翻訳元の文，連声前の文を翻訳先の文としてみなし，sentencepiece [3] で前処理を行ってから，Attention を加えた seq2seq モデルを用いて連声後文から連声前文への翻訳モデルを作成した．この研究は，Krishna et al. (2017) のデータセットから約 10 万文をデータセットとしている．

seq2seq はその名の通り，入力系列 (sequence) を受け取り出力系列 (sequence) を返す．長い系列の入出力と相性が悪い seq2seq に，系列の各要素の特徴を保持する Attention メカニズムを加えることで，長い系列の入出力も高精度で扱うことができる．ここでは，入力として翻訳元の単語が順序を持ち並ぶ文を受け取り，出力として翻訳先の単語が順序を持ち並ぶ文を返す．sentencepiece は日本語などのテキスト処理において必須となる分かち書きに用いられるモデルである．以下のように，スペース (_ で表現) も 1 文字と見なして文そのものを分割した上で，それぞれ区切られたものを新しい文における単語と見なす．

(1) (original) putraṁ vaṁśakaraṁ rāma nṛ-pasaṁnidhau

(2) (sentencepiece) _putraṁ _vaṁś akar aṁ_rāma

[*1] そもそも分割の必要性があるという判断をしなければいけない．

[*2] わかりやすさのために，短母音を ă で表記する．

[*3] *ihā́sti* = *ihá asti*, *índrā́* = *índra á* など．

©2022 Information Processing Society of Japan

nr̥pa saṁnidh au (Reddy et al. [2] 原文ママ)

2.2 RNN + CNN / Classic + Vedic

Hellwig and Nehrdich [5] は，文字単位の n-gram による回帰型ニューラルネットワーク (RNN) と畳み込みニューラルネットワーク (CNN) を組みあわせたネットワークを構築した．さらに先の Reddy et al.[2] の研究を含むいくつかの手法を比較した．それにより，CNN と RNN を組み合わせたネットワークによる連声の解除は，最も精度が高いことを示した．

この研究で用いられたテキストには，古典サンスクリット語のテキストとヴェーダ語のテキストが含まれている．しかしながら，ヴェーダ語のテキストからはアクセントが取り除かれている．そのため，アクセント記号の除去されていないヴェーダ語のテキストには，この連声解除モデルを適用し難い．単語のどの位置にアクセントがあるかによって意味が異なるヴェーダ語では，アクセントのない状態からアクセントを復元するようなことはできない．

2.3 Transformer / Vedic

塚越 [6] は，いずれの先行研究もアクセントを持つヴェーダ語のテキストを扱わない中，アクセント記号が含まれるヴェーダ語のテキストを対象とした．この研究は Reddy et al. [2] の翻訳問題への変換という手法を採用し，当時 (から今でもなお) state-of-the-art の Transformer モデルを用いて連声解除モデルを作成した．Transformer は，RNN とともに用いられてきた Attention だけを取り出し，(self) Attention 層の組み合わせのみから成る．種々の課題に対して高精度な結果を出しており，一般的に翻訳問題においても高い精度が得られる．

Hellwig and Nehrdich [5] は確かにヴェーダ文献をデータセットに含めているものの，アクセント記号を除去している．それゆえ，前節 (2.2) でも述べたように，ヴェーダ語が持つ，古典サンスクリットと大きく異なる特徴の一つである単語のアクセントが考慮されていない．1.1 節で見たように，アクセントの有無によって単純に得られる候補の数が増加する．このことから，古典サンスクリット用の連声解除モデルを，そのままヴェーダ語に応用することは難しい．一方で，ヴェーダ語を対象にした塚越 [6] は，アクセントを考慮に入れているものの，当時最も精度が良い Reddy et al. [2] にわずかながら及ばない．また，同時期に発表された Hellwig and Nehrdich [5] の手法は，それらより高精度の連声解除を実現している．ただし，それぞれの研究はアクセントのあるデータセットとアクセントのないデータセットを用いているため，単純な比較は不適切である．

アクセントが保持されているヴェーダ語のテキストに対して連声解除を行えるのは，この研究において作成されたモデル以外にない．

3. n-gram 翻訳

ヴェーダ語に特有のアクセントは連声の規則にも関係する．そのため，前節 (2節) で挙げた古典サンスクリットのみを対象とする研究 [2]，あるいはヴェーダ語も対象としながらもアクセント表記を消去した研究 [5] では，アクセント表記のなされたヴェーダ語テキストを適切に処理できない．アクセント表記のなされたヴェーダ語のテキストを処理できる唯一の研究である塚越 [6] の研究は，精度の観点で課題が残されている．そこで，本研究はアクセントのあるヴェーダ語のテキストを対象とした連声解除の手法に有効な前処理の手順を提案し，それが既存の研究結果に勝ることを示す．

3.1 手法

本研究では，関連研究 (2 節) と同様に，連声解除前のテキストから連声解除後のテキストへ翻訳するタスクを扱う．用いるテキストは，ヴェーダ文献の 1 つ『リグ・ヴェーダ』である．『リグ・ヴェーダ』は，連声解除前のテキストと連声解除後のテキストが既に存在しており，教師あり学習に最適な文献である．テキストはそのまま用いるのではなく，『リグ・ヴェーダ』の各行を文字単位の n-gram に置き換えるという単純な前処理を施す．$n = 2, \cdots, 6$ のテキストを用意し，いずれの n-gram 翻訳が最も優れているか判断する．翻訳は，Transformer モデルを用い，トークナイザー及び翻訳のモデルを作成した．

©2022 Information Processing Society of Japan

それぞれの手順の詳述は以下の節で行う．

3.1.1 文字単位 n-gram

この実験では，与えられた文を文字単位 n-gram に並べたものを新しく「文」として扱う．文字単位 n-gram とは，文を n 文字で分割することである．1.1 節で例示した *y/o+hatv/āhim* (= *yó hatvā́him*)[*4] という文で n-gram を具体的に示す．n = 3 とすると，連声後の *y/o+hatv/āhim* と連声前の *y/aḥ hatv/ā /ahim* それぞれに対して，

(3) y/o /o+ o+h +ha hat atv tv/ v/ā /āh āhi him (連声後)
y/a /aḥ aḥ+ ḥ+h +ha hat atv tv/ v/ā /ā+ ā+/ +/a /ah ahi him (連声前)

を得る．(1) に示した sentencepiece の文のように，新たに区切られたセグメントを単語と見なしている．

本手法は Kitagawa and Komachi [7] やそれに従う Hellwig and Nehrdich [5] の n-gram 手法とは異なる．それらの手法において，文字に対応する入力層に，その文字を起点/終点とした n-gram 文字列も同時に埋め込む (図1)．つまり，1 つのモデルの中で決められた範囲内の n すべての n-gram を用いて，それを入力とする．一方で，本手法は，n-gram に並べられた文字列をそれぞれを 1 つの単語のように扱う (例文 (2)，図 2)．1 つのモデルに対して 1 つの n を選んで n-gram の文を作るのである．

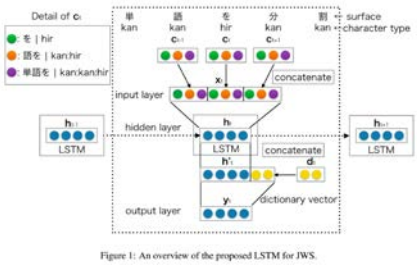

Figure 1: An overview of the proposed LSTM for JWS.

図1　Kitagawa and Komachi (2017) の埋め込み

[*4] 後述の n-gram テキストにおける見やすさのためにアキュート記号はスラッシュ，スペースはプラスで表記する．

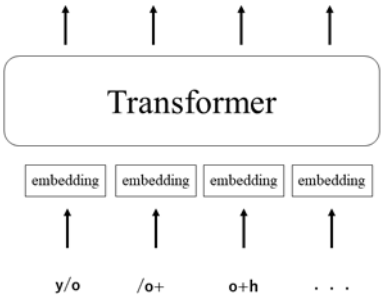

図2　本手法の n-gram の埋め込み

このような文字単位 n-gram に分割された新しい文からなるテキストを n = 2, 3, 4, 5, 6 で作成した．また，文字転写以外のテキスト加工を施さない「素」のテキスト ([6] と同一のテキスト) も用意した．これら 6 種のテキストを後述 (3.1.2節) の学習用テキストに対して作成し，後述 (3.1.3節) のモデルによって翻訳を行う．

3.1.2 学習用テキスト

本実験で用いるテキストは，『リグ・ヴェーダ』の電子テキスト (Martínez García and Gippert [8]) である．『リグ・ヴェーダ』は，連声規則適用後のテキストであるサンヒター・パータと，連声規則適用前のテキストであるパダ・パータが伝えられている．パダ・パータの作者は，オリジナルの『リグ・ヴェーダ』とも言えるサンヒター・パータの創作者らよりも後代の者である．また，連声解除は必ずしも一意に行われないことから，パダ・パータの連声解除形は複数ある解釈の 1 つであることに注意せねばならない．それを踏まえても，連声前後のテキストの組があるということは，教師あり学習にきわめて好都合である．それゆえ，約 10000 詩節から構成される『リグ・ヴェーダ』は，ヴェーダ語のテキストを学習する際に最適な文献である．

3.1.3 モデル

機械学習，特に深層学習によるテキスト分析は様々な課題を高精度でこなせている．翻訳もその一つであり，それそのものの是非をここでは議論しないが，実用されている場面を目にすることは少なくない．

©2022 Information Processing Society of Japan

さて，本実験では，連声解除の問題を連声後のテキスト (サンヒター・パータ) から連声前のテキスト (パダ・パータ) への翻訳と捉え，広く活用される翻訳モデルをここに導入することで問題解決を図る．ここでは Transformer モデルを用いた手法を採用する．Tensorflow ライブラリを用いて連声解除の課題を設定したスクリプトを作成し，深層学習に適したコンピュータ上で実験を実施する．

作成したトークナイザー `https://colab.research.google.com/drive/19A1BNyVxdcfqU5Kvgc_t729RLYZMw3Pu?usp=sharing`，翻訳モデル `https://colab.research.google.com/drive/1SlzrrxFmGI0LLz7Td9NFvOgQut0GJ5dg?usp=sharing` は公開されている．

3.2 結果

表 1 には，n-gram に分割しないそのままのテキストに対して Transformer で学習したモデル (Word)，n-gram (n = 2, 3, 4, 5, 6) に分割したテキストそれぞれに対して Transformer で学習したモデルの計 6 手法に対して Precision (適合率), Recall (再現率), F1 (Precision と Recall の調和平均) を示す．参考として表 2 に，Hellwig and Nehrdich (2018) が提案する CNN と RNN の合成モデルほかそれまでの研究が提示した精度を示す．

表1　n-gram テキストの精度

	Precision	Recall	F1 score
Word	0.950	0.966	0.958
2-gram	0.878	0.883	0.881
3-gram	0.948	0.959	0.954
4-gram	0.958	0.972	0.965
5-gram	**0.960**	**0.977**	**0.968**
6-gram	0.953	0.972	0.962

Reddy et al. [2] は，タイトル ‘Building a Word Segmenter for Sanskrit Overnight’ にもあるように学習時間の短さを強調する．言語資源が時々刻々と増加する現代語では，学習も含めた処理時間が問題となる．しかし，古典語であるサンスクリットは言語資源の量がほぼ一定であり，実用上，処理時間は問題にならない．そのため，ここでは各モデルの学習および評価にかかった時間は注目されない．

表2　Hellwig and Nehrdich (2018) の Table 3 抜粋

	Precision	Recall	F1 score
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
$\text{rcNN}_{\text{short}}{}^{\text{split}}$	94.6	94.8	96.7

4. 結論

単語の並ぶ文に対して特別な処理を行わない学習 ([6]) に比べて，n-gram に分割するという前処理を適用したあとの学習のほうが，より正確に課題をこなす場合がある．特に n = 5 のとき，最も精度が高い (表 1)．

この n の値は連声の音環境に関係すると考えられる．連声が起こる音環境の中で最も文字列が多いものは，-/as+X- (> -/a+X- or -/o+X-; / はアクセント，+ はスペース，X は有声音を示す任意の記号) である．この文字列はちょうど 5 文字 (/, a, s, +, X) であり，n-gram 文は n = 5 のとき，過不足なくこの連続を含む．それゆえに n = 5 が最も高精度を出すと考えられる．このことから，アクセントを持たない古典サンスクリットのテキストに対しては n = 4 のときに最も精度が高いと予想できる．

本実験は『リグ・ヴェーダ』のみを用いていており，一方 Hellwig and Nehrdich [5] はヴェーダと古典どちらも含むサンスクリットのテキストを用いているため，結果の単純な比較はできない．そのことに注意しながらも，本実験の結果 (表 1) を表 2 と比べてみると，わずかながら 5-gram + Transformer モデルの精度が高い．Helliwig and Nehrdich [5] と同一のデータセットに対して n-gram 分割と Transformer を

©2022 Information Processing Society of Japan

用いることで，先の仮説の検証と同時にモデルの比較ができるようになる．そうであっても，そもそも Precision, Recall ともに十分な精度をもつ 5-gram + Transformer は，ヴェーダ語の連声解除において有効に活用されうる．

精度の向上が 1 % 前後しかなくともテキスト処理においては有意義である．例えば，データベースの作成を考えると，人力で誤りを訂正する労力を減らす利点を持つ．古典文献に比べれば，ヴェーダ文献はテキストの総量が少ないながらも，ヴェーダ語を対象とした言語処理は発展途上である．前処理としての連声解除は，ヴェーダ語 × 自然言語処理の発展に大きく寄与する．

また，この n-gram 分割の手法はヴェーダ語の連声解除問題に限らず，アクセントを持たない古典サンスクリットや他の言語の単語分割問題にも応用できる．その時には，各言語における関係する音連続の最大文字列数あるいはそれに近い値での n-gram によって高い精度を得ると期待される．

謝辞

本研究は JSPS 科研費 20J23373 の助成を受けたものです．

参考文献

[1] Macdonell, A. A.: A Vedic Grammar, Verlag von Karl J. Trübner (1910).

[2] Reddy, V., Krishna, A., Sharma V. D., et al.: Building a Word Segmenter for Sanskrit Overnight, Proc. *LREC 2018*, pp. 1666-1671, ELRA (2018).

[3] Schuster, M. and Nakajima, K.: Japanese and Korean voice search, Proc. *ICASSP 2012*, pp. 5149-5152, IEEE (2012).

[4] Krishna, A., Satuluri, P., and Goyal, P.: A dataset for Sanskrit word segmentation, *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pp. 105-114, ACL (2017)

[5] Hellwig, O. and Nehrdich, S.: Sanskrit Word Segmentation Using Character-level Recurrent and Convolutional Neural Networks, Proc. *EMNLP 2018*, pp. 2754-2763, ACL (2018).

[6] 塚越柚季: Transformer モデルを用いた機械学習によるサンスクリットの連声解除，じんもんこん 2018 論文集，pp. 9–14, (2018)

[7] Kitagawa, Y. and Komachi, M.: Long short-term memory for Japanese word segmentation, Proc. *PACLIC 32*, pp. 279-288, ACL (2018)

[8] Martínez García, F. J. and Gippert, J.: Plain text retrieval, Thesaurus Indogermanischer Text- und Sprachmaterialien (オンライン), 入手先 <http://titus.fkidg1.uni-frankfurt.de/private/texte/indica/vedica/rv/pp/rvarpp.txt> (参照 2022-10-17).

©2022 Information Processing Society of Japan