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[*1]

塚越 柚季 (東京大学大学院人文社会系研究科)

サンスクリットの文中の単語に自動で形態情報を付与するのは容易ではない．文中の語形 (主に語尾) は，連声規則により主に後続する語の語頭の音によって変化する．このため，手を加えていないテキストに対する形態情報の付与が難しい．そこで連声規則が適用されている原典テキストから，連声規則前の形の単語の連続に戻すことが必要である．

時間を要するが，語彙や形態の情報を元に連声を解除することは可能である．一方でそのような情報なしに Attention メカニズム + sequence to sequence モデルを用いて短時間に高精度で連声の解除にも成功している．

本研究では Attention のみを使って高い精度を出すことができる Transformer モデルを用いて，高精度な連声の解除を行った．

Sanskrit Sandhi splitter by machine learning using Transformer

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

It is difficult to automatically give morphological information to words in a Sanskrit sentence. Sandhi rules change the word forms, especially final sounds, depending on the initial sound of the following word. This makes automatic glossing of untokenized Sanskrit sentences difficult. Therefore, it is necessary to restore individual words from original sentences combined by the sandhi rules.

Although it takes a long time, it is possible to split sandhi by vocabulary and/or morphological information. Alternatively, there is another approach that does not need vocabulary or morphological information but splits sandhi with less time, but higher accuracy.

This research was performed to split Sanskrit sandhi by the Transformer model which gives high accuracy using only Attention.

1. 連声

サンスクリットの連声は，文中において特定の音環境で隣接した語が結合する場合と隣接部の音が変化する場合がある．この連声によって変化した形はテキストにもそのまま反映される．例えば，*sū́ryasyeva* は *sū́ryasya* 「太陽. 属格単数」，*iva* 「ように」と 2 つの語が結合して現れた形である．また *stómo vasiṣṭhās* は *stómas* 「賞賛. 主格単数」，*vasiṣṭhās* 「Vasiṣṭha. 呼格単数」の 2 語から成るが，1 語目の語末が連声によって *-as* から *-o* に変化している．

連声規則が適用される条件は決まっており，その変化後の形は多くの場合一意に定まる．しかし連声が適用された文中の語に対して，音韻のみを手がかりにそれを連声適用前の形に戻すのは容易ではない．例えば，語末の *i, ī* は，後続する語の始まりが *i, ī* 以外の母音ならば，どちらも *y* になるという規則のように，連声前の音と連声後の音が多対一の対応を示す場合が多いからである．

上記の例のような場合でも，形態や意味の情報があれば，ありうる変化形を元に連声前の形に戻すことができる．例として，*jánitry ajījanat* 「生み出す者 (である女神) が生み出した」は *jánitrī* + *ajījanat* であって，*jánitri* + *ajījanat* ではない．これは語幹 *jánitrī-* というものがあって，これは主格単数形で *jánitrī* となり，呼格単数形が *jánitri* だが，文脈から主格が適

[*1] 本研究における実験では科学研究費補助金 (基盤研究 C) 18K00524 を利用した．

当であることと文頭以外では呼格はアクセントを持たないことためである．また，*jánitri* という活用形を持つ語幹も存在しないためである．このようにして，与えられた語形から活用のもととなる語幹を推測し，その活用形が把握することが可能ならば，普通は連声適用前の形に戻すことができる．

2. 『リグ・ヴェーダ』

サンスクリットは，ヴェーダ文献で用いられるサンスクリットと，それ以降の時代に用いられるサンスクリットに分けて考えられる．讃歌集『リグ・ヴェーダ』はヴェーダ文献の中でも最も古い．そのため『リグ・ヴェーダ』で用いられる言語には，それ以外のヴェーダ文献に用いられる言語とも，ヴェーダ期以降の言語とも異なる点が見られる．中でも，本研究が取り組む連声規則は，時代・文献ごとに異なり，『リグ・ヴェーダ』はさらにその中でも巻 (後述) ごとにも連声規則が異なる．

『リグ・ヴェーダ』は全 10 巻から構成される．各巻はいくつかの詩節を持ち，その詩節は基本的に 4 行 1 組とする詩連から成る．連声規則は一部の詩行をまたいで適用される．

『リグ・ヴェーダ』には，連声が適用されたまま伝わる「サンヒターパータ」と，連声が適用される前の語に戻した形の「パダパータ」とが存在する．

3. seq2seq + Attention

sequence to sequence (seq2seq) は系列を入力として系列を出力する深層学習のモデルである [6]．例えば，これを機械翻訳に用いると，英語の文 (単語の系列) を入力としてフランス語の文 (単語の系列) を出力するというような学習が行える．学習時にソース (英語) の各要素を 1 つ前の要素と合わせながら順にベクトルに変換するエンコーダと，ターゲット (フランス語) の各要素をエンコーダの要素も含んだ 1 つ前の要素と合わせながらベクトルに変換し，そのベクトルから系列を出力するデコーダとから成る (図1).

seq2seq はその構造上，系列が長いほど系列の後ろの方におけるベクトルに系列の始めの方の影響が現れにくい．Attention メカニズムは，そのような難点を補い，長い系列に対しても良い精度を出すことができる [3]．上記のような英語-フランス語翻訳の場合，学習時に Attention 層においてソースとターゲットのそれぞれの系列の各要素 (単語) の関連度も合わせて学習する (図2)．Attention を用いないときは，固定次元のベクトルの中に圧縮された特徴のみを参照していたのが，この Attention 層の参照によりデコード時に系列の後ろの方でも系列の始めの方との関連度から直接その特徴を参照することができるようになる．

4. 関連研究

Reddy らは Attention を加えた seq2seq を用いた連声の解除は，語彙・形態を手がかりにした分析よりも早く高精度に連声適用前に復元することに成功した [4]．またこの分析は，語彙や形態などの言語学的情報を用いないものである．

その方法では入力系列を連声後のテキストとし出力系列を連声前のテキストとしている．データは，Digital Corpus of Sanskrit の中の Sanskrit Word Segmentation Dataset [*2] を用い，その中から 107000 文を学習用のデータとし，別の 4200 文を試験用のデータとして用いた．ただし，テキストは学習のために sentencepiece モデル [5] によって加工が施されている．

彼らの結果は，既存の最良の手法に比べ Precision, Recall, F1-Score (後述) のいずれにおいても点数が上である．

5. Transformer モデル

Attention メカニズムは seq2seq が使われるようなリカレントニューラルネットワークなどで用いられてきた．その中，Transformer という Attention のみに依ったモデルが提案された [8]．複雑な構造をしたリカレントネットワークと結びつけられた Attention から離れて，単純に Attention のみを用いる構造によって，学習データが大きい必要がなく，学習にかかる

[*2] https://zenodo.org/record/803508#.WTuKbSa9UUs

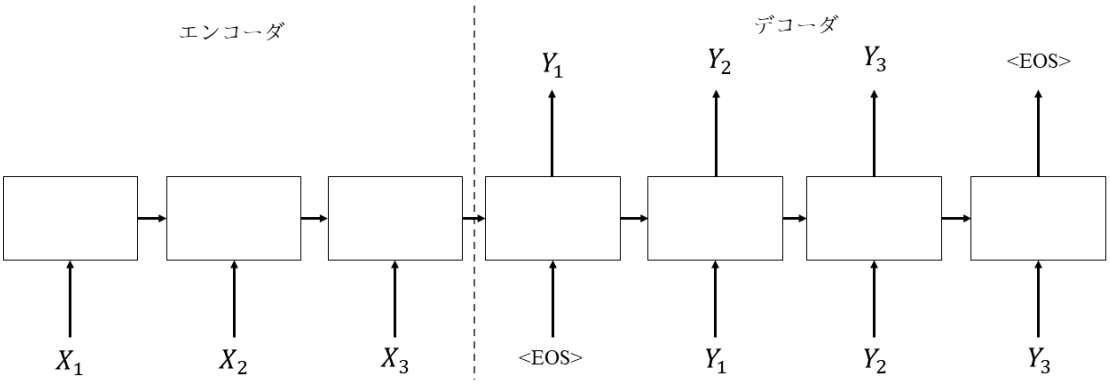

図1　seq2seq モデル

X_i がソース文字列の各要素，Y_j がターゲット文字列の各要素．<EOS> は文の終わりを示す．ここでは 1 層の構造で図示したが多層構造も存在する．

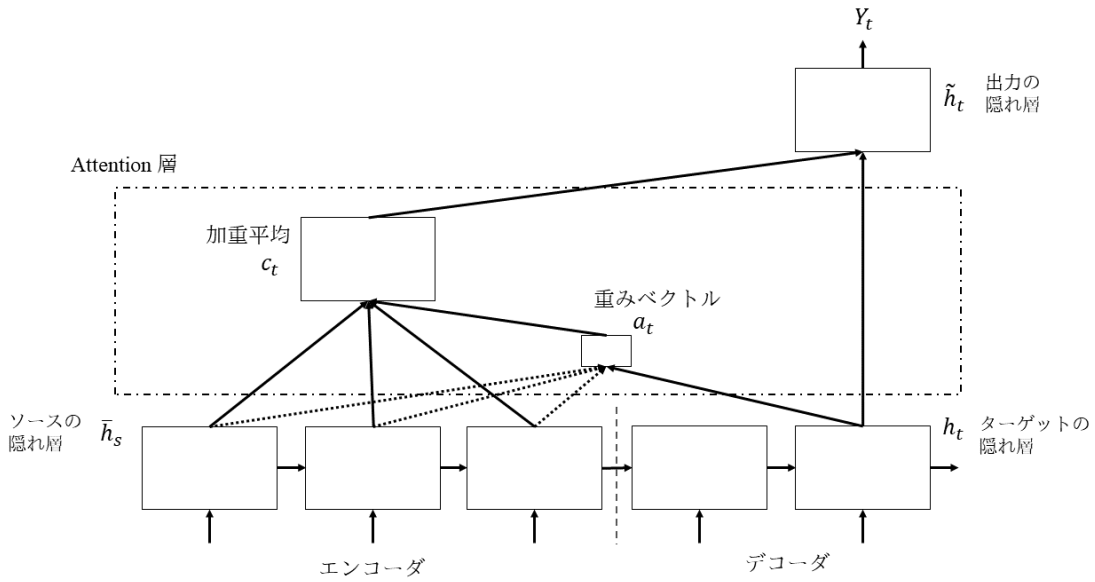

図2　Attention メカニズム

h_s がソース要素の隠れ層ベクトル，h_t がターゲット要素の隠れ層ベクトル．a_t (=$a_t(s)$) はある s, t に対してそれらの関連度を表す重みベクトル．c_t は重み a_t と h_t の積を全 t について足し上げた加重平均．a_t, c_t の層が Attention 層．$\tilde{h}_t$ は Attention 層で計算された c_t とターゲットの隠れ層ベクトル h_t から得られる，出力の隠れ層ベクトル．

時間が減った上，精度も既存のモデルより高いという成果が得られた．

6. 手法

Tensor2Tensor (T2T) という深層学習モデルやデータセットのライブラリの中に Transformer モデルが収められており，問題およびデータを自身で設定したスクリプトを用いて [7]，それに従って学習を行った[*3]．

本研究では，Reddy ら [4] と同様に，入力として連声後のテキストである「サンヒターパータ」を，出力として連声前のテキストである「パダパータ」を用いて，Transformer モデルによって学習を行った．学習用のデータの作成のためにデジタルテキスト [1] から「サンヒターパータ」と「パダパータ」を抽出，整形し，翻字方式を Tokunaga-Fujii 方式に変更した．使用したデジタルテキストは，各詩行ごとに改行を施している．しかしながら，連声規則は詩行をまたいで適用され，また，本実験で行った学習はデータの 1 行 1 行を対応させるものなので，学習および試験データは 1 行が 1 つの詩節となるような整形を施した．さらに全 10551 文を無作為に二分し，それぞれ訓練データと評価用の試験データとした．

また比較のため，『リグ・ヴェーダ』の一部を学習し，もう一部に対して学習が良好か評価するのに加えて，Reddy ら [4] の用いたデータを使った学習も行った．彼らの実験では，扱うデータが予め sentencepiece モデル [5] で加工されており，その手順が不透明である．そのため，彼らの手法によって『リグ・ヴェーダ』の連声解除をするのは容易ではないことから，学習モデルを固定した比較の代わりにデータを固定した比較を行った．

7. 実験結果

学習の評価として表1に BLEU および Precision, Recall, F1-Score のスコアをあげる．Transf:RV は本実験で用いた，モデルが Transformer，データが『リグ・ヴェーダ』であり，比較として，Transf:DCS はモデルが Transformer，データが Reddy らの用いたデータ (DCS) である．参考として Reddy らの seq2seq + Attention (atSeq2Seq) のスコア (原文ママ) も載せる．

表1 評価スコア

モデル	BLEU	Precision	Recall	F1
Transf:RV	0.9370	0.8842	0.8788	0.8815
Transf:DCS	0.9603	0.8000	0.6667	0.7273
atSeq2Seq	–	0.9077	0.903	0.9053

BLEU は機械翻訳の評価において主流の評価方法である．その値が 1 に近いほど機械が生み出した訳と予め用意しておいた参照用の訳とが類似している，つまり学習が良好に行われたということである．本実験においては，試験データである「サンヒターパータ」を機械が学習によって連声解除した文と，既存の「パダパータ」の文とを比べた．

Precision，Recall は情報検索や分類問題などで用いられる評価方法であり，いずれも 1 に近いほど精度が良い．連声の解除の場合には，機械が学習によって生み出した連声解除の文 (という集合を G，参照用の「パダパータ」の文 (という集合) を O としたとき，Precision P は，集合 G に占める，G と O の積集合，すなわち機械の学習による連声解除と実際の正しい連声解除の一致した箇所の集合 $G \cap O$ の割合である．一方で Recall R は，母数を機械が生み出した連声解除の文 (という集合) G ではなく，参照用の「パダパータ」の文 (という集合) O に置き換えたものである．F1-Score F_1 は，Precision P と Recall R との調和平均であり，Precision, Recall の各々の値よりもこの値が重視される．実際はその集合の要素数で計算するため以下のように求められる．

$$P = \frac{|G \cap O|}{|G|}, R = \frac{|G \cap O|}{|O|}$$

$$F_1 = \frac{2}{P^{-1} + O^{-1}} = \frac{2PO}{P + O}$$

[*3] 使用したスクリプトおよびデータは `https://github.com/Yuzki/sandhi` にある．

本実験においては利用した GPU は Geforce GTX 1080Ti (11GB メモリ，3584 コア) であり，処理にかかった時間は，Transformer を用いた『リグ・ヴェーダ』の連声解除の学習には 11 時間 50 分程度だった．DCS の連声解除の学習には，10 時間 30 分程度かかった．

8. 考察

Transformer モデルは Attention メカニズムを用いた seq2seq モデルよりも良い成果を出すとされているが，本実験による学習の結果は，F1-Score において Reddy らの結果よりわずかに劣る．この原因として，扱ったデータの違いが考えられる．Reddy らはヴェーダ期を除く文献の中から 107000 文をデータとして，4200 文を試験に用い，残りのうち約 7500 文を除いて全て訓練に用いた．一方で本実験では『リグ・ヴェーダ』10551 文全ての中から約半分の 5277 文を訓練に，残り 5274 文を試験に用いた．各実験の 1 文の長さは異なるものの，訓練データの量および試験データの量が大きく異なる．このことから本実験の訓練が十分ではなかった可能性が考えられる．しかし，訓練の量を考慮した評価の指標がないため定量的な記述はできないが，Transformer モデルによってわずかな訓練データによる学習であっても大量の訓練データによる学習に及びうる成果を出せることが示された．

Transformer を用いた『リグ・ヴェーダ』の連声解除の学習にかかった時間は 12 時間弱，DCS の学習には 10 時間 30 分程である一方で，Reddy らが行った Attention + seq2seq を用いた DCS の学習は 11 時間 40 分かかった．両者の機械環境が同一ではないため単純に要する時間を比較しても，参考程度にしかならないが，同一データの DCS に関しては本実験のほうが処理時間が長い．ここで，両者の環境を考えると，本実験で使用した GPU は Geforce GTX 1080Ti (11GB メモリ，3584 コア)，Reddy らの使用した GPU は Titan X (12GB メモリ，3584 コア) である．利用するモデルが違うのではあるが，メモリ数が劣る本実験環境において処理時間が長いというのは想定できることである．しかし，一般に高精度とされる Transformer を用いたほうが，Attntion + seq2seq を用いるよりも，精度が下がっている．スクリプトを本実験に特化したものにはしていないことや，データ量が多くとも過学習をしていしまう恐れがあることなど原因となりうることは考えられるが，このような精度の下落の原因は明らかではない．

学習に要する準備に関して言えば，手を施していないデータをそのまま使用した本実験のほうが，Reddy らの研究のように sentencepiece を用いて一度学習データ，試験データを変換するよりも，準備に必要な全体の時間は少ない．

9. 結論

Transformer モデルを用いて，『リグ・ヴェーダ』をデータとした連声解除の学習は，既存の手法にわずかながら精度に劣るように見えるものの，少ない量の学習データでそのような精度を出すことができる．さらに，扱うテキストデータを予め加工せず，そのままのデータを用いて学習し，学習に必要な時間も大きく変わらないことから，費用対効果が高い．

Reddy らは，最も古い時代のヴェーダ期を除く，それ以降の様々な時代の文献を含むコーパスの文をデータとして用いた．一方で，本研究の実験では，その文献内で時代の幅があるもののヴェーダ文献の 1 つである『リグ・ヴェーダ』という 1 つの文献に限ったデータを用いた．ヴェーダ期 (さらに古い時代の『リグ・ヴェーダ』) のテキストを用いた本研究と，ヴェーダ期よりも新しい時代のテキストを用いた Reddy らの研究はどちらも，時代・文献が異なることによる連声の差異への適応に難色を示しうる．しかしながら，本研究で用いた Transformer モデルは学習データが小さい量で良くデータの加工も不必要という利点から，学習データの作成がより簡便になる．このことによって，時代・文献ごとの適応可能性が十分高いことが見込まれる．

サンスクリットの連声規則の中には，時代・文献によって異なる規則が存在する．そのため，複数の文献を訓練データとした場合と，1 つの文献を訓練データ

とした場合のいずれも連声を解除する際の精度に影響を及ぼしうる．そこで，話者ごとに ID を付した学習も可能 [2] であることから，この話者を文献に置き換えることで時代・文献の差異も含んだ 1 つのネットワークの構築が可能になると考えられる．

サンスクリットの連声を高精度で解除できることでその形態情報や語彙情報の付与が自動で行いやすくなる．これによってサンスクリット文献を用いた量的処理が容易になることが期待される．

参考文献

1) 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 2018-08-31).
2) Li, L., Galley, M., Brockett, C., et al.: A Persona-Based Neural Conversation Model, (2016).
3) Luong, M.-T., Pham, H., Manning, C. D.: Effective Approaches to Attention-based Neural Machine Translation, (2015).
4) Reddy, V., Krishna, A., Sharma V. D., et al.: Building a Word Segmenter for Sanskrit Overnight, (2018).
5) Schuster, M. and Nakajima, K.: Japanese and korean voice search, (2012).
6) Sutskever, I., Vinyal, O., Le Q. V.: Sequence to Sequence Learning with Neural Networks, (2014).
7) T2T: train on your own data, 入手先 <https://tensorflow.github.io/tensor2tensor/new_problem.html> (参照 2018-08-31).
8) Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need, (2017).