連載

5分で分かる!? 有名論文ナナメ読み

中澤敏明（東京大学）

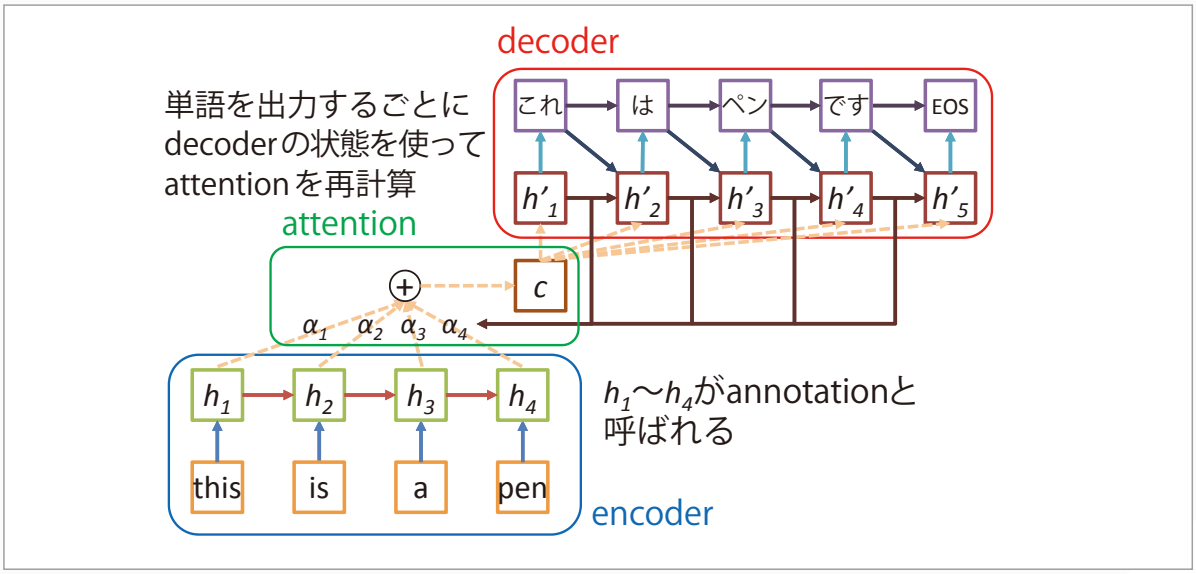
Ashish Vaswani et al. : Attention Is All You Need

Advances in Neural Information Processing Systems, pp.5998-6008（2017）

ニューラル機械翻訳の変遷

今回紹介する論文はタイトルだけを見ると何の論文だかまったく見当もつかないと思われるが，中身は自然言語処理，特にニューラル機械翻訳（NMT）に関する論文である．中身の説明に入る前に，これまでのニューラル機械翻訳について振り返る．

end-to-end のニューラル機械翻訳が初めて提案されたのは文献 1）の論文である．ここで提案されたのは encoder と呼ばれる Recurrent Neural Network（RNN）と，decoder と呼ばれる別の RNN を繋げたモデルであった（図 -1 の attention を除いた部分）．encoder は入力文を 1 単語ずつ読み込み（実際には各単語は分散表現と呼ばれるベクトルで表現されている），入力文の情報全体を 1 つのベクトル（図 -1 の h_4）に圧縮する役割を担っている．decoder は encode された入力文のベクトルを最初の入力として受け取り（図 -1 で h_4 が h'_1 に入力される），翻訳文の最初の単語（図 -1 の「これ」）を出力する．以降は 1 つ前に出力した単語を decoder の次の入力とすることで，次の出力単語を得る．これを文の終わりを表す EOS（End-Of-Sentence）が出力されるまで繰り返すことで，任意の長さの出力を得ることができる．この論文で提案された encoder-decoder モデルにより，入力と出力で長さや順序が異なる問題や，入力と出力で構造が異なる問題などでも end-to-end の学習が行えるようになり，その後さまざまな分野で使われている．



◆図 -1　attention の計算例（EOS は文末を表す記号）

このモデルの最大の欠点は，どんな長さの入力であっても固定長のベクトルに圧縮しているため，入力が長くなるにつれてすべての情報を保持することができなくなくなり，精度が悪化するという点である．この欠点を解決したのが文献2）の論文である．ここで提案されたのは，encoderの各単語の隠れ層のベクトルをすべて保存しておき（annotationと呼ばれる．図-1のh_1からh_4），decoderがこれらの情報を参照しながら次の出力単語を決定するというモデルであった．図-1の緑で囲った部分に示すように，各入力単語はattentionと呼ばれる重み（図-1のα_1からα_4）付きで参照され，その和であるc（コンテキストベクトルと呼ばれる）を使って次の出力単語を決定する．attentionは1単語出力するたびにdecoderの隠れ層の状態（h'_1からh'_4）と入力の各単語の隠れ層のベクトル（h_1からh_4）から再計算される．このattentionにより長い入力での精度の悪化を防ぐことに成功し，attention付きのRNN enc-decモデルはNMTのスタンダードな方法として定着した．現在のGoogle翻訳もこのモデルをベースとしている．なおattentionという考えはこの論文以前にすでに提案されていたが，NMTに適用したのはこの論文が初である．

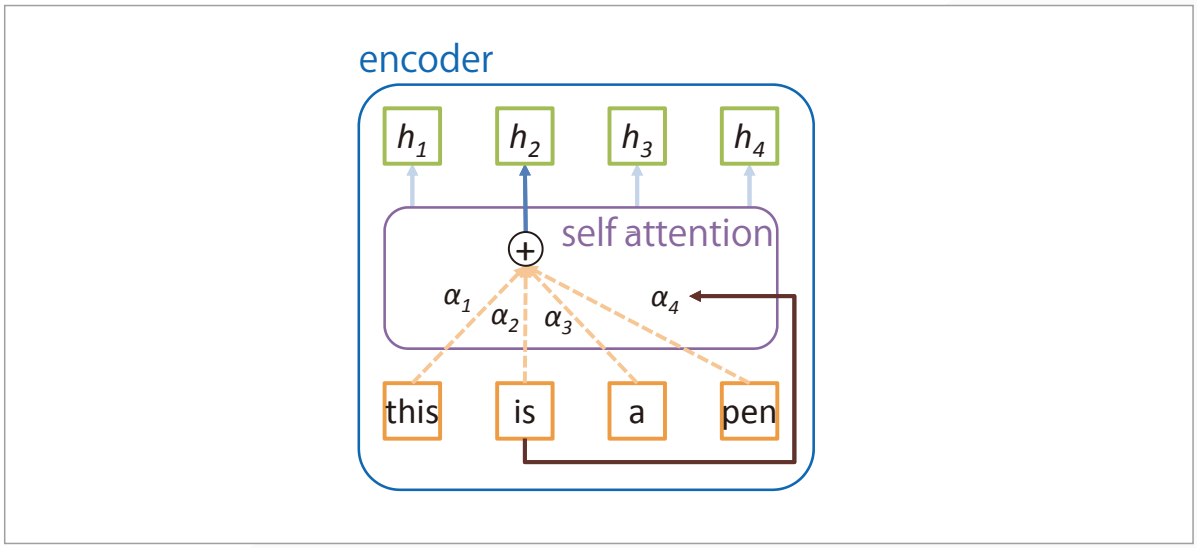
この後しばらくはRNN enc-dec＋attentionが盛んに研究されてきたが，RNNは1単語ずつ読み込む必要があるため，処理を並列化できず学習に時間がかかるという欠点がある．そこでRNNの代わりにConvolutional Neural Network（CNN）を利用する方法が提案された[3]．CNNにより処理を並列化することができるため，学習が高速に行える一方で，CNNでは局所的な情報しか利用できないため，遠い位置にある単語間の関係を考慮したい場合にはCNNの層を深くする必要があるという欠点がある．

RNNもCNNも使わないNMT

さてここまでくれば，今回紹介する論文のタイトルの意味も推測できるかもしれない．この論文ではencoder，decoderともRNNもCNNも用いず，attention機構とFeed Forward Neural Network（FFNN）のみを用いるTransformerというモデルを提案している．Transformerでは入力文をencodeするために，self-attentionという仕組みを利用する．通常のattentionは次に出力する単語を決める際に入力文のどの単語に注目するかを考慮するものであるが，self-attentionは入力文の各単語のannotationを作り出すために用いられる．入力中のある単語（仮にX_iとする）について考えると，同じ入力の中で注目する単語の重みを計算したものがself-attentionであり，入力の各単語のベクトル表現をself-attentionの値で重み付き和を取ったものが，その文内での単語X_iのannotationとなる．同様にして他の入力単語についてもannotationが計算されるが，self-attentionの値はannotationを計算する単語ごとに変化する．図-2にTransformerのencoderにおいて，“is”のannotationを計算している例を示す．ほかの単語についても同様にannotationが計算でき，これらの計算はRNNとは違って独立に実行できるため，並列化可能である．

self-attentionは入力文内で自分と関係の深そうな単語への重みが大きくなるように学習され，結果として文の構造のようなものを反映することができる．たとえば代名詞の場合はその先行詞にattentionされたり，動詞と補語が離れた位置にあっても動詞から補語にattentionされたりする．decoderについても同様にself-attentionを用いるが，翻訳文の生成は先頭から順に行われるため，未来の情報が先に使われないように工夫されている．

各単語のattentionの計算は並列に行えるため高速に動作する上，CNNとは違って各単語は文内の



◆図 -2　Transformer の encoder の例

ほかのすべての単語との関連を考慮することができる．実験では計算量がほかのモデルより 1 桁以上少ないにもかかわらず，翻訳の精度は報告された時点で最高精度を達成したことが示されている．また Transformer を英語の構文解析に適用した結果も報告されており，最高精度には届かなかったが非常に高精度で行えることが示されている．つまり Transformer が文の構造を捉えているということである．

2018 年 6 月には Transformer をさまざまな自然言語処理タスクに応用したところ，多くのタスクで最高精度を達成したという報告がなされた[4]．今後も Transformer を元にした研究が数多く報告されることだろう．

参考文献

1) Sutskever, I. et al. : Sequence to Sequence Learning with Neural Networks, NIPS2014, pp.3104-3112.
2) Bahdanau, D. et al. : Neural Machine Translation by Jointly Learning to Align and Translate, ICLR2015.
3) Gehring, J. et al. : Convolutional Sequence to Sequence Learning, ICML2017.
4) https://blog.openai.com/language-unsupervised/

（2018 年 8 月 10 日受付）

中澤敏明（正会員）nakazawa@logos.t.u-tokyo.ac.jp

2010 年京都大学大学院博士課程修了．博士（情報学）．京都大学特定助教，科学技術振興機構研究員などを経て，2018 年より東京大学特任講師．自然言語処理，特に機械翻訳に関する研究に従事．