係り受け構造に基づく Attention の制約を用いた NMT

出口 祥之 † 田村 晃裕 ‡ 二宮 崇 ‡
† 愛媛大学 工学部 情報工学科，‡ 愛媛大学 大学院理工学研究科 電子情報工学専攻
† deguchi@ai.cs.ehime-u.ac.jp
‡{tamura, ninomiya}@cs.ehime-u.ac.jp

1 はじめに

近年，自然言語処理の多くのタスクにおいて，ニューラルネットワークが活用されている．機械翻訳の分野においてもその有効性が示されており，その中でも，Transformer [1] というモデルが RNN や CNN ベースのモデルの翻訳精度を上回り，注目を浴びている．特に，Transformer の特徴の一つである self-attention は文内における単語間の関連の強さを考慮することができ，これを用いることで，機械翻訳のみならず，言語モデルの獲得，Semantic Role Labeling (SRL) など，様々なタスクにおいて精度の向上に寄与してきた．

これまでに統計的機械翻訳や RNN を用いた Sequence to Sequence のニューラル機械翻訳モデルなどでは文構造を考慮することで機械翻訳の性能を改善してきたが，Transformer モデルでは文構造を陽に考慮したものはない．そこで，本研究では，Transformer モデルで係り受け構造を考慮することで翻訳性能の改善を試みる．

Transformer モデルで係り受け構造を考慮するという研究は機械翻訳においてはないが，SRL の分野においては存在し，Linguistically-Informed Self-Attention (LISA) [2] では self-attention に対して係り受け構造に基づく制約を与えて学習させることで State-of-the-Art の性能を達成している．

本稿では LISA により提案された手法を Transformer の機械翻訳モデルに適用することで翻訳性能の改善を試みる．具体的には，self-attention の一部に対して係り受け構造に基づく制約を与えることで，係り受け関係を考慮した単語間の関連性を学習させ，翻訳時にはその self-attention を用いることで，係り受け関係を考慮した翻訳を行う．ここで，LISA は Transformer エンコーダのみを使うモデルであるが，提案の機械翻訳モデルはエンコーダとデコーダのそれぞれの self-attention で係り受け構造に基づく制約を与える点が LISA とは異なる．

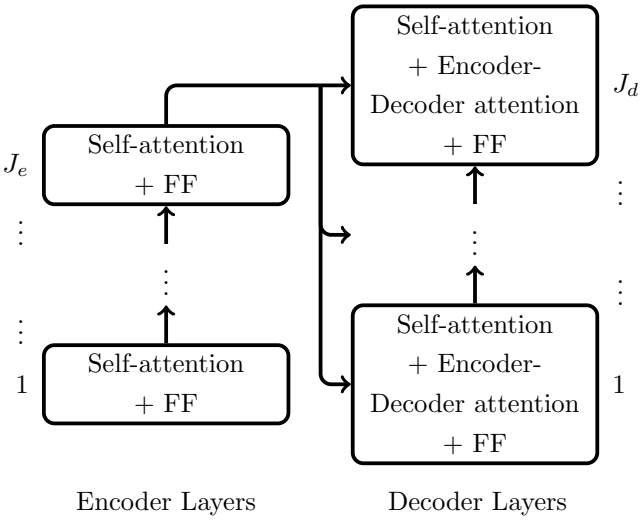

図 1: Transformer モデル

Asian Scientific Paper Excerpt Corpus (ASPEC) データ [3] を用いた日英翻訳の評価実験において，提案モデルと従来の係り受け構造を考慮しない Transformer モデルを比較し，係り受け構造を考慮することで BLEU が 0.49 向上することを確認した．

2 Transformer

本節では，提案モデルの基礎となる Transformer [1] を説明する．

Transformer は，入力された原言語の単語列 $X(=x_1, x_2, \ldots, x_{n_{enc}})^T$ をエンコードする Transformer エンコーダ (以下，エンコーダ) と目的言語の単語列 $Y(=y_1, y_2, \ldots, y_{n_{dec}})^T$ を生成する Transformer デコーダ (以下，デコーダ) を組み合わせたニューラル機械翻訳モデルである．エンコーダとデコーダはそれぞれ，図 1 のようにエンコーダレイヤを J_e レイヤ，デコーダレイヤを J_d レイヤずつ積み重ねたものである．

Transformer は入力文に対して並列に処理するため単語の位置情報を考慮できない．そこで，単語が入力されたら，その単語の位置情報を位置エンコーディング (positional encoding) により考慮する．具体的には，単語埋め込みの次元数を d とすると，位置エンコーディングの行列 P は

$$P_{(pos,2i)} = sin(pos/10000^{2i/d}) \quad (1)$$

$$P_{(pos,2i+1)} = cos(pos/10000^{2i/d}) \quad (2)$$

と表される．なお，pos は単語の位置，i は埋め込み成分の次元である．そして，式 (1)，(2) によって計算された位置エンコーディングの行列 P を単語の埋め込み行列に加算したものが，エンコーダ及びデコーダの入力となる．

エンコーダレイヤとデコーダレイヤの第 j レイヤの出力をそれぞれ $S_{enc}^{(j)}$，$S_{dec}^{(j)}$，エンコーダレイヤとデコーダレイヤ内の attention レイヤをそれぞれ $Attention_{enc}^{(j)}(\cdot)$，$Attention_{dec}^{(j)}(\cdot)$ とし，また，Layer Normalization を $LN(\cdot)$ とすると，エンコーダの第 j レイヤの出力 $S_{enc}^{(j)}$ は，

$$S_{enc}^{(j)} = LN(S_{enc}^{(j-1)} + Attention_{enc}^{(j)}(S_{enc}^{(j-1)})) \quad (3)$$

と表され，デコーダの第 j レイヤの出力 $S_{dec}^{(j)}$ は，

$$S_{dec}^{(j)} = LN(S_{dec}^{(j-1)} + Attention_{dec}^{(j)}(S_{dec}^{(j-1)})) \quad (4)$$

と表される．なお，$S_{enc}^{(0)}$，$S_{dec}^{(0)}$ はそれぞれ，エンコーダ，デコーダへの入力を表す．ここで，attention レイヤ $Attention_{enc}^{(j)}(\cdot)$ は，multi-head self-attention，feed-foward ネットワークを順に接続したレイヤであり，$Attention_{dec}^{(j)}(\cdot)$ は，masked multi-head self-attention，multi-head encoder-decoder attention，feed-foward ネットワークの順で接続したレイヤである．ただし，attention レイヤ内は残差接続によって接続されている．残差接続では，ある関数 $\mathcal{F}$ の出力に対して，入力したものが足し込まれ，残差接続の結果は $x + \mathcal{F}(x)$ となる．

multi-head attention は単語間の関連の強さを考慮するための機構であり，単語の埋め込み次元を n_{head} 個の $d_{head} = d/n_{head}$ 次元の部分空間に射影し，それぞれの部分空間で attention の計算を行う．self-attention では，直前のレイヤの出力を $S^{(j-1)} \in \mathbb{R}^{n \times d}$(ただし，$n$ はエンコーダ/デコーダの入力系列長) とすると，$W_h^{Q^{(j)}} \in \mathbb{R}^{d \times d_{head}}$，$W_h^{K^{(j)}} \in \mathbb{R}^{d \times d_{head}}$，$W_h^{V^{(j)}} \in \mathbb{R}^{d \times d_{head}}$ によって d 次元の $S^{(j-1)}$ を $Q_h^{(j)}$，$K_h^{(j)}$，$V_h^{(j)}$ の d_{head} 次元の部分空間に射影する (なお，$1 \leq h \leq n_{head}$)．デコーダで用いられる encoder-decoder attention では，直前のレイヤの出力を $Q_h^{(j)}$ に，エンコーダの出力を $K_h^{(j)}$，$V_h^{(j)}$ の部分空間に射影する．射影後，次の式によって各部分空間で単語間の関連の強さを表す行列を得る．

$$A_h^{(j)} = softmax(d_{head}^{-0.5} Q_h^{(j)} {K_h^{(j)}}^T) \quad (5)$$

この $A_h^{(j)}$ に対して $V_h^{(j)}$ を掛け合わせることで，トークンに対して単語間の関連の強さを重みとする荷重和による表現 $M_h^{(j)}$ を得ることができる．

$$M_h^{(j)} = A_h^{(j)} V_h^{(j)} \quad (6)$$

最後に，各部分空間の $M_{1,2,\ldots,n_{head}}^{(j)}$ を次式のように結合させ，単語の埋め込み次元に線形変換する．

$$M^{(j)} = W^{M^{(j)}}[M_1^{(j)}; M_2^{(j)}; \ldots; M_{n_{head}}^{(j)}] \quad (7)$$

ただし，$W^{M^{(j)}} \in \mathbb{R}^{d \times d}$ である．

なお，目的言語の出力文は，デコーダの J_d レイヤの出力を単語次元に線形変換した後，$softmax$ 関数をかけることで算出される $P(Y \mid X)$ に基づき生成する．

3 提案手法

本稿で提案するモデルでは，LISA [2] の手法を用いて self-attention の一部で係り受け解析を行う．LISA は SRL において，Transformer Encoder に係り受け解析を行う syntactically-informed self-attention を導入することで，それまでのモデルよりも大きく精度を向上させたモデルである．本稿では，図 2 のように，エンコーダの第 p_e，デコーダの第 p_d レイヤに syntactically-informed self-attention を導入し，機械翻訳の精度向上を図る．係り受け解析をする方法としては，LISA 同様，Dozat らの Deep Biaffine parser [4] の手法を用いる．

3.1 Syntactically-Informed Self-Attention

エンコーダもしくはデコーダの第 p レイヤに syntactically-informed self-attention を導入する場合，multihead self-attention を以下のように変更する．

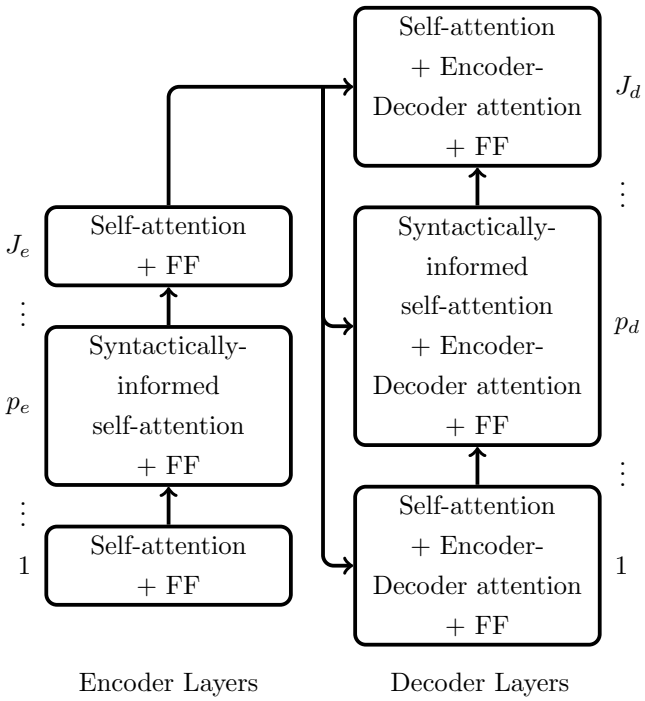

図 2: 提案モデル

1. 直前のレイヤの出力 $S^{(p-1)}$ を次式によって線形変換する．

$$Q_{parse} = S^{(p-1)}W^{Q_{parse}} \quad (8)$$
$$K_{parse} = S^{(p-1)}W^{K_{parse}} \quad (9)$$
$$V_{parse} = S^{(p-1)}W^{V_{parse}} \quad (10)$$

なお，$W^{Q_{parse}}$，$W^{K_{parse}}$，$W^{V_{parse}}$ はそれぞれ単語埋め込みの次元 d から multihead attention の各部分空間の次元 $d_{head} = d/n_{head}$ に変換する $d \times d_{head}$ の行列である．

2. 各単語間の係り受け関係らしさを表す attention weights (A_{parse}) を Biaffine 変換 [4] によって計算する．$U^{(1)} \in \mathbb{R}^{d_{head} \times d_{head}}$，$\mathbf{u} \in \mathbb{R}^{d_{head}}$ とすると，次式によって表される．

$$A_{parse} = softmax(Q_{parse}U^{(1)}K_{parse}^{T} + Q_{parse}U^{(2)}) \quad (11)$$

ただし，$U^{(2)} = \overbrace{(\mathbf{u} \ldots \mathbf{u})}^{n \text{ 個}}$.

3. A_{parse} と V_{parse} を掛け合わることで，係り受け構造を考慮した表現を荷重和として取り出す．

$$M_{parse} = A_{parse}V_{parse} \quad (12)$$

4. 通常の multihead attention の $M_h^{(p)}$ の内の 1 つを M_{parse} によって置換する．

5. 通常の multihead attention と同様に，各部分空間を結合させ単語埋め込みの次元 d に線形変換する．

$$M^{(p)} = W^{M^{(p)}}[M_1^{(p)}; M_2^{(p)}; \ldots; M_{parse}; \ldots; M_{n_{head}}^{(p)}] \quad (13)$$

ただし，$W^{M^{(p)}} \in \mathbb{R}^{d \times d}$ である．通常の multihead attention は式 (7) で表されるのに対し，syntactically-informed self-attention では式 (13) のように表され，$M_h^{(p)}$ の内の一つが M_{parse} で置換されていることがわかる．

3.2 学習方法

本稿で提案するモデルでは，LISA に倣って機械翻訳と係り受け解析を同時に学習する．

機械翻訳の誤差を e^{tokens}，エンコーダ側の係り受け解析の誤差を e_{enc}^{parse}，デコーダ側の係り受け解析の誤差を e_{dec}^{parse} とすると，モデル全体の誤差を

$$e^{tokens} + \lambda_{enc}e_{enc}^{parse} + \lambda_{dec}e_{dec}^{parse} \quad (14)$$

とする．ただし，λ_{enc}，λ_{dec} は正の定数である．この誤差関数を最小化するように提案モデルを学習する．e^{tokens} と e_{enc}^{parse}，e_{dec}^{parse} は，ラベル平滑化交差エントロピー [5] を用いて算出する．また，式 (11) の A_{parse} においてトークン t が係り先 q を指す尤度を $A_{parse}[t, q]$，q が t の係り先であることを $q = head(t)$ とすると，

$$P(q = head(t) \mid X) = A_{parse}[t, q] \quad (15)$$

と表される．ただし，係り先の無いものは自分自身を指すようにする．

4 実験

4.1 実験設定

本実験では，提案手法の有効性を確かめるため，syntactically-informed self-attention を組み込んだ提案手法モデルと組み込まなかった従来の Transformer モデルの精度を比較する．

表 1: 日英翻訳での精度比較

モデル	BLEU
ベースライン	21.16
提案手法	21.65

データは，Workshop on Asian Translation で用いられた ASPEC[3] の日英対訳コーパスを使用した．英語の単語分割には Stanford Tokenizer[1]を用い，日本語の単語分割には KyTea を用いた．すべてのアルファベットは小文字化した．

訓練データは train-1.txt から抜き出した 50 単語以下からなる対訳文対の内，前方の 10 万文対を用いた．語彙は単語の出現頻度が 2 未満のものを <unk> トークンに置き換え，出現頻度が 2 以上のものを使用した．

英語側の係り受け解析には Stanford Dependencies[2] を用い，日本語の係り受け解析には EDA 係り受け解析器[3]を用いた．

モデルのパラメータの最適化は Adam [6] を用いて行った．学習率の調整や Adam のハイパーパラメータの設定は，*warmup_steps* を 8000 に設定した以外は，Vaswani ら [1] の設定に従った．エンコーダレイヤとデコーダレイヤはそれぞれ 6 層ずつスタックし，multi-head attention は 8 つの部分空間に分割し，単語の埋め込み次元は 512 次元に設定した．また，syntactically-informed self-attention はエンコーダ，デコーダともに第 4 レイヤ目に導入した．

モデルの誤差について，ラベル平滑化交差エントロピーの ϵ_{ls} [5] は $\epsilon_{ls} = 0.1$ とした．また，係り受け解析の誤差の影響度をコントロールするパラメータ λ_{enc}，λ_{enc} はそれぞれ 0.5 とした．

ミニバッチの大きさは 100 に設定し，エポック数 50 まで学習させ，開発データ (dev.txt) に対して最も性能の良いモデルを評価した．評価はテストデータ (test.txt) でグリーディ法によって生成した文に対して行った．

4.2 結果

提案モデルとベースラインの Transformer モデルの翻訳性能を表 1 に示す．なお，翻訳精度は BLEU で評価した．表 1 より，self-attention に係り受け構造に基づく制約を用いることで，ベースラインとなる Transformer モデルよりも BLEU が 0.49 上がったことが確認できる．

[1]https://nlp.stanford.edu/software/tokenizer.html
[2]https://nlp.stanford.edu/software/stanford-dependencies.html
[3]http://www.ar.media.kyoto-u.ac.jp/tool/EDA/

5 おわりに

本研究では，Transformer に syntactically-informed self-attention を組み込むことで，Transformer による機械翻訳で係り受け構造を考慮する手法を提案した．そして，ASPEC の英日翻訳タスクの評価を通じて，提案手法により係り受け構造を考慮することで BLEU が 0.49 改善できることを確認した．提案モデルは，LISA [2] 同様，推論時に，外部の係り受け解析器を用いて解析した係り受け構造も翻訳で活用することが可能である．今後は，外部の係り受け解析結果との統合を行い，さらなる機械翻訳の精度向上を目指したい．

6 謝辞

本研究成果は，国立研究開発法人情報通信研究機構の委託研究により得られたものである．また，本研究の一部は JSPS 科研費 25280084 及び 18K18110 の助成を受けたものである．ここに謝意を表する．

参考文献

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in NIPS 30*.

[2] E. Strubell, P. Verga, D. Andor, D. Weiss, and A. McCallum. Linguistically-informed self-attention for semantic role labeling. In *Proc. of EMNLP 2018*, 2018.

[3] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. Aspec: Asian scientific paper excerpt corpus. In *Proc. of LREC 2016*, 2016.

[4] T. Dozat and C. D. Manning. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*, 2016.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on CVPR*.

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.