局所的依存構造をSelf-Attentionにより考慮する翻訳文生成

露木 浩章　　小川 哲司　　小林 哲則　　林 良彦
早稲田大学理工学術院
tsuyuki@pcl.cs.waseda.ac.jp

1 はじめに

本研究では，未来の単語 (未生成の単語) を参照できない条件下における局所的依存構造を考慮した翻訳文生成手法を提案し，その有効性を評価する．なお，局所的依存構造とは，注目単語と直接依存関係にある単語からなる部分木を指す．

依存構造を考慮した機械翻訳手法が多数提案されている．特に，翻訳文の生成タスクと依存構造解析タスクとの同時学習によって目的言語側の依存構造を考慮する場合，文法的破綻の少ない翻訳文の生成が期待できる [1]．しかし，単語を生成しながら生成済み単語列に対して依存構造解析をするため，原言語側で依存構造をする場合と比べて，未来の単語列を参照できないという難しさがある．近年の研究に Transformer に対して依存構造を考慮する Syntactically-informed Self-Attention [2] を導入した手法 [3] があるが，注目単語に対する親単語[1]を推定する手法のため，親単語が未生成の場合にその依存構造を考慮できていない．

本研究では，依存構造を考慮した翻訳文生成を目的として，依存構造解析と翻訳文生成の同時学習手法を提案する．親単語だけでなく，子単語も同時推定する Syntactically-informed Self-Attention の学習によって，生成単語の局所的依存構造を考慮した翻訳文生成を行う．また，単語間の依存関係の有無だけでなく，依存関係の種類を考慮するため，依存関係の種類も同時推定する手法を導入する．実験の結果，Multi30K コーパスを用いた独英翻訳タスクにおいて 0.59，ASPEC コーパスを用いた日英翻訳タスクにおいて 0.46 の BLEU スコア改善を確認した．さらに翻訳例を比較した結果，提案手法は依存構造の正しい翻訳文生成に寄与することを確認した．提案手法は機械翻訳と同じ Seq2Seq モデルを利用する文章要約タスクや対話応答タスク，言語モデルを利用するタスク等適用可能範囲の広い手法である．

[1]木構造上において直上にある単語を親単語，直下にある単語を子単語と呼ぶ．

2 関連研究

機械翻訳タスクに対して依存構造を考慮したモデルを提案した研究について述べる．考慮される依存構造は学習済み解析器から得られたものである．

2.1 原言語側で依存構造を考慮した研究

Transformer を利用した機械翻訳モデルにおいて，原言語側の依存構造を考慮する手法は，翻訳と同時に入力文の依存構造解析を行う手法 [4]，依存構文木上における単語間の距離を入力する手法 [5], 依存構造木を系列で表現し，単語列と組み合わせて入力する手法 [6] がある．特に文献 [4] で使われている Syntactically-informed Self-Attention (SynSA) は依存構造解析手法の 1 つである [7]．Multi-Head Self-Attention の 1Head 目の Attention が注目単語に対する親単語を推定するように学習する．このときの Attention は，注目単語の子単語へのなりやすさと単語ペアの依存関係へのなりやすさの和，2 つの入力ベクトルの Biaffine 変換によって親単語推定の尤度を計算する．SynSA を利用した依存構造解析手法は，意味役割付与や機械読解タスクとの同時学習によって各タスクの最高性能を達成している [2, 8]．本研究では SynSA を利用するが，目的言語側の依存構造に焦点を当てて手法の検証をするため，原言語側の依存構造を考慮しない．

2.2 目的言語側で依存構造を考慮した研究

機械翻訳タスクに対して，目的言語側の依存構造を考慮する研究は 4 つに大別できる．1 つ目は依存構造木を系列に変換する前処理を行った後に，Seq2Seq モデルを適用する手法である [1]．単語列中の適切な位置に”{”と”}”，依存関係の種類を表す関係ラベルを挿入し，文全体の依存構造木を深さ優先の規則に沿った系列で表現する．依存構造木を表す系列を Seq2Seq モデルの学習データとして利用することによって，翻訳文中の重複を抑制し長文を含む学習データに対してよ

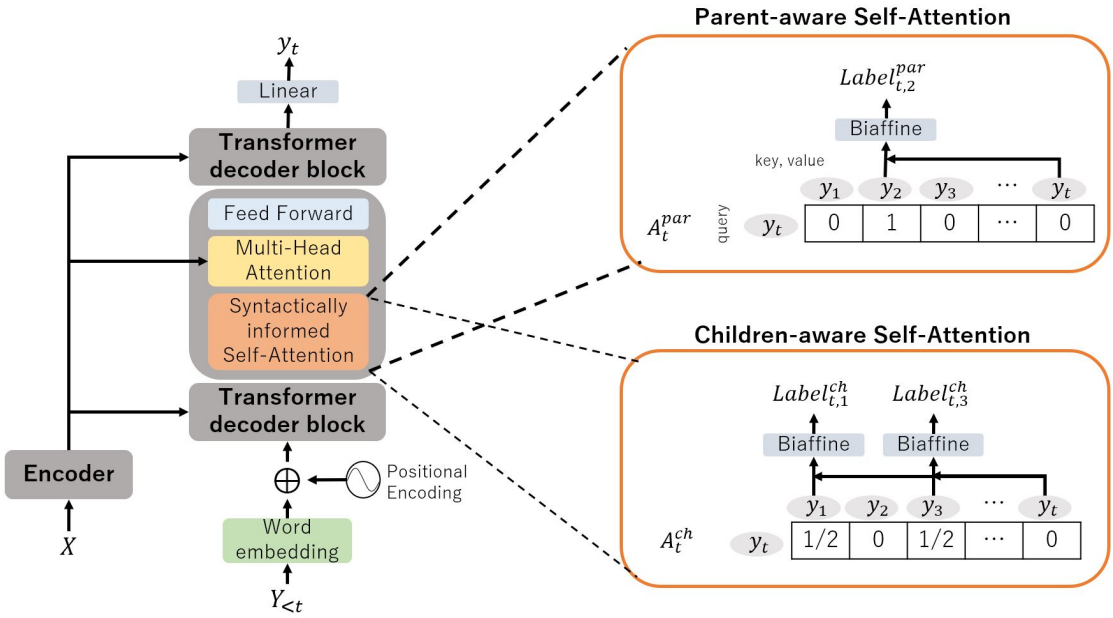

図 1: 提案手法の全体像

り顕著な性能向上があったと報告されている．シンプルでパラメータも増えない手法であるが，依存構造木を系列で表現する前処理の結果，系列の長さが 2 倍以上の長さになるという性質がある．

2 つ目は木構造をそのままモデル化し学習する手法である [9]．句構造木上のノードの親子関係を表す RNN と兄弟関係を表す RNN を用いて Top-down，Left-to-Right に構文木を生成・学習することによって，性能向上すると報告されている．一方，CFG ルールを利用した文献 [9] の類似手法を依存構造木に対して適用した結果，通常の Seq2Seq よりも性能が低下したという報告もある [10]．

3 つ目は shift-reduce 操作を用いた遷移型の依存構造解析手法を単語生成と同時に学習する手法である [11]．shift 操作によって単語はスタックとバッファに追加され，reduce 操作によってスタック中の単語間の依存関係が決定される．提案された NMT+RNNG では，入力文エンコーダの出力と 3 つの LSTM (バッファから生成済み単語列をエンコードする LSTM，生成済み shift-reduce アクション系列をエンコードする LSTM，スタックから解析済み句構造をエンコードする LSTM) の出力から次の shift-reduce 操作を推定する．しかし，shift-reduce 操作による依存構造解析は依存関係が交差する構造 (日本語の場合，話し言葉に頻出する．例：「これが私は正しいと思う」) を表現できず，複数の LSTM を利用するために必要パラメータ数が大きくなるという問題がある．

4 つ目は Tranformer の Self-Attention を利用する手法である [3]．前述の SynSA をデコーダに適用することによって，系列長はそのままに必要パラメータ数を増やさず，交差する依存構造を表現可能である．しかし，文献 [3] では，生成済みの単語列に対して，注目単語の親単語を推定する SynSA を適用していたため，親単語が未生成の場合に依存関係を考慮できていなかった．本研究では子単語を推定する SynSA を導入する事によってこの問題へ対処する．加えて依存関係の種類も同時推定する層を加えることによって，より豊富な依存構造情報を考慮する．

3 提案

図 1 に提案手法の全体像を示す．目的言語側の依存構造解析のための層を，親単語を推定する Self-Attention と子単語を推定する Self-Attention に分けて導入し，それぞれを Parent-aware Self-Attention (PSA) と Children-aware Self-Attention (CSA) と呼ぶ．m 番目の Tranformer デコーダブロック中の Multi-Head Self-Attention のうち 2 つの head を PSA と CSA に置き換えた．つまり，PSA と CSA で計算された Attention は上層の Transformer デコーダブロックへの入力の計算に使われるだけでなく，それ自体が親単語推定タスクと子単語推定タスクの尤度として扱われる．予備実験の結果，次の単語の推定前にその単語の依存構造解析をする層を適用すると，現在の単語について

依存構造解析する場合と比べて，高い性能となった．したがって，本研究では単語推定前にその依存構造解析の層を導入する．

3.1 局所的依存構造解析

本節では Parent-aware Self-Attention について述べる．Children-aware Self-Attention についても同様に学習する．

入力文を X，t-1 番目の単語まで翻訳済みの単語列を $Y_{<t} = \{y_0, y_1 ... y_{t-1}\}$ とする．また，m 番目の層において，y_{t-1} の key, value, query をそれぞれ K_t, V_t, Q_t とする．A_t^{par} は y_t に対して，各生成済み単語 $Y_{\leq t}$ が親単語である尤度を表す Attention 行列である．y_t に対する親単語の尤度は以下の式 1 で計算される．

$$P(A_t^{par} \mid X, Y_{<t}) = \text{BiaffineAttn}(Q_{0<s\leq t}, K_t) \quad (1)$$

文献 [7] に習い，Biaffine Attention を適用した．図 1 の右上方に示す行列には，y_2 が y_t の親単語である場合の例を示す．この行列は親単語との Attention の値が 1，それ以外は 0 の行列である．$Y_{<t}$ 中に親単語がない場合は自分自身を親単語とした．推定した attention 行列は親単語推定の尤度として使われるだけでなく，$V_{0<s<t}$ と内積を計算した後，上層に渡される．

同様にして親単語との依存関係の種類について推定した．t 番目の単語とその親単語との依存関係の種類を $Label_t^{par}$ としたとき，Biaffine 変換を適用した尤度の計算式を以下の式 2 に示す．

$$P(Label_t^{par} \mid X, Y_{<t}, A_t^{par}) = \text{Biaffine}(V_{0<s\leq t}, A_t^{par}) \quad (2)$$

図 1 の右上方に示す例では，y_t と親単語 y_2 との関係を推定している．$Y_{<t}$ 中に親単語が存在しない場合は，関係ラベルを $< no\_parent >$ とした．

3.2 目的関数

式 3 に提案手法の目的関数を示す．λ_{pa},λ_{ca}, λ_{pL}, λ_{cl} は各項の重みを表すハイパーパラメータである．依存関係の種類を推定する際に，正しい親子関係 (依存関係) に対してその種類を学習させるため，学習時は正解の $A_{t,G}^{par}$ と $A_{t,G}^{ch}$ をモデルに与えた．翻訳文推定や親単語，子単語推定の際には，モデルに依存構造の情報は与えていない．翻訳文推定時に，モデルのパラメータが増えたり，依存構造推定のエラーが伝搬することを防ぐためである．

$$\begin{aligned} J(\theta) = \sum_t & \log P(y_t \mid X, Y_{<t}) \\ & + \lambda_{pa} \log P(A_t^{par} \mid X, Y_{<t}) \\ & + \lambda_{ca} \log P(A_t^{ch} \mid X, Y_{<t}) \\ & + \lambda_{pl} \log P(Label_t^{par} \mid X, Y_{<t}, A_{t,G}^{par}) \\ & + \lambda_{cl} \log P(Label_t^{ch} \mid X, Y_{<t}, A_{t,G}^{ch}) \end{aligned} \quad (3)$$

4 実験設定

日英翻訳タスク Asian Scientific Paper Except Corpus (ASPEC)[2]の日英コーパスを使用した．文献 [11, 3] と同様に，学習データとして 50 単語以下の対訳文ペアを，事前に付けられた類似度スコアの高い順に上位 10 万ペアだけ抽出し，使用した．日本語の単語分割には KyTea[3]を，英語の単語分割，依存構造解析には Stanford Tokenizer と Stanford Parser[4]を利用した．学習データ中に出現した依存構造関係を表すラベルは 45 種であった．

独英翻訳タスク Multi30K [5]の独英コーパスを使用した．学習データは 2.9 万文ペアであった．ドイツ語の単語分割，英語の単語分割，依存構造解析には spaCy[6]を利用した．学習データ中に出現した存構造関係を表すラベルは 47 種であった．

ハイパーパラメータ 文献 [3] に合わせてデコーダレイヤ全 6 層の内，4 層目に SynSA (PSA と CSA) を導入，100 文ずつ 50 エポックまで学習し，検証データに対して最も BLEU スコアの高いモデルをテストデータで評価した．その他の設定も文献 [3] に従った．検証データを使用した探索に基づき，λ_{pa},λ_{ca}, λ_{pL}, λ_{cl} はそれぞれ 0.2, 0.2, 0.3, 0.3 とした．

5 実験結果

表 1 に，日英翻訳と独英翻訳における各モデルの BLEU スコアを示す．比較手法は NMT+RNNG [11], Trans.+DBSA [3]，Tranformer である．提案手法は Tranformer に対して，独英翻訳タスクで 0.59，日英翻訳タスクで 0.46 ポイントの BLEU スコア改善を示した．また，提案手法は原言語側と目的言語側，両方の

[2]http://lotus.kuee.kyoto-u.ac.jp/ASPEC/
[3]http://www.phontron.com/kytea/index-ja.html
[4]2https://nlp.stanford.edu/software/
[5]http://www.statmt.org/wmt16/multimodal-task.html
[6]https://spacy.io/

表 2: ASPEC コーパスにおいて生成された翻訳文の例.

入力文：土壌の硬度は，作物の根の生育に直接または間接的影響を与える。	
	翻訳文
正解	Soil hardness *directly or indirectly affects* the growth of plant roots .
Transformer	The hardness of the soil *directly or the indirect effect* on the growth of the root of the crop .
提案手法	The hardness of the soil *directly or indirectly affects* the root growth of crop .

表 1: BLEU スコアの比較実験結果.

モデル	De-En	Ja-En
NMT+RNNG [11]	-	18.84
Transformer [3]	-	21.16
Trans. + DBSA [3]	-	21.56
Transformer (ours)	38.20	21.17
提案手法	**38.79**	**21.63**

依存構造を SynSA によって考慮した Trans.+DBSA を上回るスコアを示した．目的言語側で子単語推定や依存関係の種類推定の学習を導入して，依存構造を考慮したことが，原言語側の依存構造を考慮することよりも性能改善に大きく寄与したいえる．

表 2 に ASPEC コーパスを利用した実験における Tranformer と提案モデルそれぞれから生成された翻訳文の例を示す．Transformer では，入力文の「直接または間接的影響を与える」という箇所が”directly or the indirect effect”と翻訳された．これに対して提案手法では，”directly or indirectly affects”と翻訳された．”directly or”まで翻訳した段階で，次に生成する単語の親単語は or であり，直前の directly と並列の関係にあることを考慮できていることから．正しい依存構造の翻訳文が生成されている．

6 おわりに

本研究では，未来の単語 (未生成の単語) を参照できない条件下における依存構造を考慮した翻訳文生成手法を提案した．親単語だけでなく子単語も推定するように学習することによって，親単語が未生成の場合でも依存構造を考慮できるようにした．また，依存関係の種類も同時推定する学習も導入し，より豊富な依存構造情報を考慮可能とした．独英翻訳と日英翻訳における実験を通して，提案手法の有効性を評価した．今後は生成された依存構造と翻訳文の比較分析や原言語側の依存構造との関係について分析を行いたい．また，利用する依存構造体系との相性，言語との相性に関する検証を進める．

参考文献

[1] An Nguyen Le, Ander Martinez, Akifumi Yoshimoto, and Yuji Matsumoto. Improving sequence to sequence neural machine translation by utilizing syntactic dependency information. In *AFNLP*, 2017.

[2] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Liuistically-informed self-attention for semantic role labeling. In *EMNLP*, 2018.

[3] 二宮崇 出口祥之, 田村晃裕. 係り受け構造に基づく attention の制約を用いた nmt. In **言語処理学会**, 2019.

[4] Chengyi Wang, Shuangzhi Wu, and Shujie Liu. Source dependency-aware transformer with supervised self-attention. 2019.

[5] Yutaro Omote, Akihiro Tamura, and Takashi Ninomiya. Dependency-based relative positional encoding for transformer nmt. In *RANLP*, 2019.

[6] Anna Currey and Kenneth Healfield. Incorporating source syntax into transformer-based neural machine translation. In *WMT*, 2019.

[7] Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. In *ICLR*, 2017.

[8] Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. Sg-net: Syntax-guided machine reading comprehension. In *AAAI*, 2020.

[9] Jetic Gu, Hassan S. Shavarani, and Anoop Sarkar. Top-down tree structured decoding with syntactic connections for neural machine translation and parsing. In *EMNLP*, 2018.

[10] Xinyi Wang, Hieu Pham, Pengcheng Yin, and Graham Neubig. A tree-based decoder for neural machine translation. In *EMNLP*, 2018.

[11] Akikio Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. Learning to parse and translate improves neural machine translation. In *ACL*, 2017.