

1 はじめに

近年、自己注意機構を用いた深層学習モデル Transformer [1] が、高い汎用性を持つことから様々なタスクで高い成果を挙げている。自己注意機構とは文中の各単語間の関連スコアを算出するもので、潜在的に統語構造情報を捉えている可能性から、これを工夫した統語構造解析手法が提案されている [2][3]。従来の文法規則のマッチングによる処理とは違い、深層学習では文をベクトルで扱うためより頑健な処理を実現できる。また、統語構造解析研究ではこれまでに構築された多くの統語構造解析結果のデータが存在するため、本研究では自己注意機構を用いた教師あり統語構造解析モデルを提案する。一方で、X (旧 Twitter) などのソーシャルメディアで記述される文は文法規則にとらわれない自由な文体のコーパスで、その構文構造について従来のルールベースの手法では解析することができない文が数多く存在する。そこで本研究では、X コーパスのように教師データの存在しない文データにも対応する教師なし統語構造解析モデルも提案する。

2 木構造エンコーダを用いた統語構造解析

通常の誤差逆伝播法ではエンコーダの中間層の情報は用いず、学習モデルの最終的な出力と教師データとの差分からなる損失に基づき、出力を修正するための誤差逆伝播によってモデルの結合荷重を調整している。それに対して、本節では中間層において教師データとの損失を捉える階層的誤差逆伝播を提案する。

Tree-Transformer Tree-Transformer は従来の全ての単語に対する網羅的な自己注意機構は採用せず、エンコーダに隣り合う単語間の依存性を捉える、Constituent Attention モジュールを導入する。Constituent Attention モジュールでは、N 個の単語を含む文の各単語に対して、文中の句または節 (constituent) の区切りを推測する確率 $a = \{a_1, \dots, a_N\}$ を生成する。確率 a を用いて、同じ constituent に属する単語同士のみで自己注意機構を作用させる制約 Constituent Priors $C_{i,j}$ が生成される。 $C_{i,j}$ は、文中の i 番目の単語と j 番目の単語が同じ constituent に含まれるかの確率である。上層に移るにつれ constituent は隣同士でマージされて徐々に大きくなり、エンコーダ内部で入力文の構文木を形作る。 $C_{i,j}$ を用いて、木構造自己注意機構は式 (1) のように求められる。

$$\text{Attention}(Q, K, V)_{tree} = C \odot \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Tree-Transformer は BERT [4] の事前学習である Masked Language Modeling (MLM) によって学習される。

2.1 提案手法

Tree-Transformer の学習に加えて、統語構造解析結果のデータを用いる半教師あり学習を行う。はじめに特定の層数のエンコーダを作成し、各層で対応する教

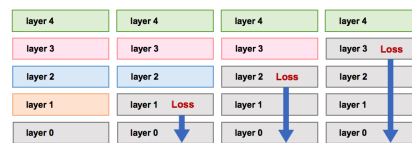


図 1: 5 層のエンコーダを作成し、教師データが 4 層の時の階層的誤差逆伝播。

師データにを用いて階層的に誤差逆伝播を行う。教師データと各層の単語間分割確率 a との数値的差分を捉え、損失関数の出力を最小化するように計算する。モデルの低層から階層的誤差逆伝播を行い、調整したパラメータを高層に向けて積み上げるように全ての層に対するパラメータを学習する (図 1)。

2.2 実験

1 文につき 3~6 単語で構文木が 3~5 階層で構成された人工データ 480 文を学習に用いる。人工データは文脈自由文法を用いて文生成を行い、Stanford Parser¹ で解析した結果を教師データとして用いた。MLM による学習と a に対する教師あり学習の損失をハイパーパラメータを α を用いて線形和に束ねる。評価は、各層で表現される a の出力と正解の構文木の a の状態を示す 0 と 1 の状態との類似度とする。全ての a が正解データと完全一致した割合を total_{acc} 、また各 a が一致した割合を total_a_{acc} とする。5 分割交差検証の最高値と最低値を除いた値を平均した結果を表 1 に示す。

損失を束ねたモデルでは、いずれも大きな変化が見られなかったが、Tree-Transformer と比べて total_a_{acc} の結果が良いことから、構文木の深さを捉えるよう学習されていることが確認できる。しかし、全体として精度が低いことから、エンコーダの出力のみを対象とした学習では精度の限界がある可能性が考えられる。

表 1: 階層的誤差逆伝播を用いた実験結果。

Model (α)	total_{acc}	total_a_{acc}
$(\alpha \times \text{loss}_{mlm} + (1-\alpha) \times \text{loss}_a)$	[%]	[%]
0.0	15.62	64.76
0.4	12.49	66.15
0.8	16.14	63.41
1.0	13.71	58.63
Tree-Transformer	19.79	32.22

3 木構造自己注意機構と Chart Decoder

代表的な統語構造解析手法として、CKY アルゴリズムがある。Kitaev ら [3] は、エンコーダに自己注意機構を、デコーダに CKY アルゴリズムを模倣した Chart Parser を用いたモデルを提案している。エンコーダの出力を学習対象としていた Tree-Transformer とは異なり、デコーダの出力を学習対象とすることにより解析された構文木を予測値とすることが可能となる。

3.1 提案手法

Kitaev らのモデルでは、自己注意機構が統語構造情報を算出することを前提としてデコーダへの入力とし

¹<https://nlp.stanford.edu/software/lex-parser.shtml>

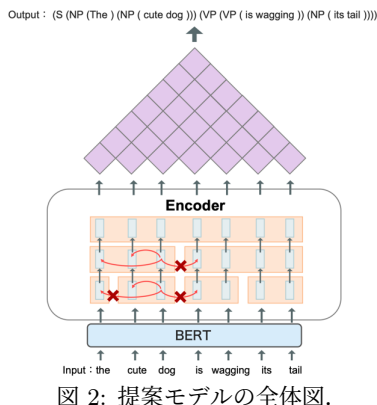


図 2: 提案モデルの全体図.

表 2: Penn Treebank での実験結果 [%].

	F_1	Complete	Tag
従来モデル ($\ell=2$)	95.75	56.22	97.45
従来モデル ($\ell=10$)	95.75	55.53	97.24
提案モデル ($\ell=2$)	95.58	55.67	97.44
提案モデル ($\ell=10$)	95.83	56.75	97.44

ている. しかし, 自己注意機構の出力には統語構造情報は明示的には反映されていないため, デコーダに用いる情報としてはまだ改善の余地がある. そこで, 確かな統語構造情報を反映するような制約を用いて抽出した特徴量をデコーダの入力とする. 図 2 に提案モデルの全体の構造を示す. エンコーダに Tree-Transformer の木構造自己注意機構を用いて, 入力文の統語構造情報を抽出し, デコーダの Chart Decoder で構文木のスコアを算出して教師あり学習を行う.

3.2 実験

実験では, Penn Treebank²の WSJ (英字経済新聞 Wall Street Journal³) コーパスを用いて, 予測される構文木が人間の作成した構文木との類似度を評価する. 提案手法の有効性を検証するため, 通常の自己注意機構を用いた Kitaev ら [3] のモデル (従来モデル) と比較を行った. エンコーダの層数を ℓ として, 表 2 に実験結果を示す. 精度の評価尺度には F_1 スコア, 再現率と適合率が共に 100% である割合 (完全一致), 品詞の正解率 (品詞) を用いた. 表より, 提案モデルが従来モデルよりも品詞以外の結果が良いことがわかる. これらのことから木構造自己注意機構は通常の自己注意機構よりも確かな統語構造情報を抽出し, デコーダでの解析を補助し, 性能改善に寄与することがわかる.

4 X コーパスの教師なし統語構造解析

多くの統語構造解析研究対象となっている英字経済新聞 (Wall Street Journal) コーパスなどと比べて X に記述される文は文法規則にとらわれない自由な文体を持つことが挙げられる. そこで本章では, 深層学習を用いて X コーパスを対象に統語構造解析を行う教師なし学習モデルを提案する.

4.1 提案手法

Recurrent Neural Network Grammar (RNNG) [5] は文とその句構造をトップダウン, また左から右に同時に生成する生成モデルの一つである. Unsupervised Recurrent Neural Network Grammars (URNNG) [6] は RNNG の, アノテーションされた構文木が必要であるという問題点に着目して開発された教師なし学習モデルである. URNNG のエンコーダには双方向 LSTM

表 3: 事前処理後の X データの例文

miss you btw where are @person and @person
nooo i dont want to get out of bed
what is ur great m0ment in cricket

が用いられているが, 自己注意機構と同様に統語構造情報を明示的に含んでいないため, 提案モデルでは木構造自己注意機構を用いる.

4.2 X コーパス

英語で記述された X コーパスを実験に用いる. コーパスに含まれる例文を表 3 に示す. 学習効率を上げるため, 文に含まれる単語数を 3~15 単語で構成されるものに限定し, それぞれ訓練データ 26,987, 評価データ 4,475 となった.

4.3 実験

X コーパスは教師データを持たないコーパスであるため, 実験では目視によって評価を行う. URNNG を用いた解析では, 3~7 単語の文においてほとんどの文が全ての単語の間で括弧が挿入されていた (図 3 上を参照). これは, 双方向 LSTM が少ない単語に対して文の依存関係を正しく抽出できていないことが考えられる. しかし提案モデルでは, 3~7 単語の文において全ての単語の間で括弧が挿入されるような解析が行われる文が URNNG と比べてかなり減少し, 多くの文で正しく構文木を解析することができていた (図 3 下を参照). 10 単語以上の長い文に対しても, 統語構造を正しく捉えているような解析がかなり増加した. これによって, 木構造自己注意機構はモデル全体のパフォーマンスを向上させたことが考えられる.

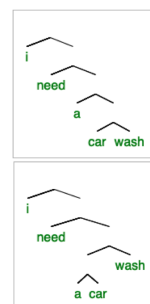


図 3: 解析結果例

5 おわりに

本稿では行った 3 つの研究について記している. 2 節では階層的誤差逆伝播を提案した. エンコーダの出力に対する教師あり学習では, 解析精度を向上させるにも限度があると感じられた. 3 節では, 木構造自己注意機構を用いたエンコーダ・デコーダモデルを提案した. 4 節では, X コーパスを解析する木構造自己注意機構を用いた教師なしモデルを提案した. 統語構造解析において木構造自己注意機構が有効であることを示した. 今後は, 教師あり学習と教師なし学習を組み合わせた半教師あり学習に取り組みたい.

参考文献

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [2] Yau-Shian Wang, Hung yi Lee, and Yun-Nung (Vivian) Chen. Tree transformer: Integrating tree structures into self-attention. In *EMNLP*, 2019.
- [3] Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In *ACL*, pp. 2676–2686, 2018.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pp. 4171–4186, 2019.
- [5] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. *CoRR*, Vol. abs/1602.07776, , 2016.
- [6] Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. In *NAACL*, 2019.

²<https://catalog.ldc.upenn.edu/LDC99T42>

³<https://www.wsj.com>