木構造 Transformer を用いた教師あり統語構造解析

成田 百花　(指導教員：小林 一郎)

1　はじめに

大規模なテキストデータから Transformer [1] を用いて事前学習をさせた汎用言語モデルである BERT [2] を用いて，目的のタスクに合わせて fine-tuning することで様々な自然言語処理課題に対して，従来の精度を大きく超える成果を挙げている．BERT は，Transformer の self-attention 機構により単語間の依存構造を捉えた言語モデルとなっており，モデル内部に潜在的に統語構造情報を含んでいると言われているが，その情報を統語構造解析に利用する研究は少ない．Wang ら [3] は，self-attention 機構を利用することにより入力文の統語構造を解析する教師なし統語構造解析器 Tree-Transformer を提案している．一方で自然言語処理においては，これまで統語構造解析の研究において構築された多くの統語解析結果のデータが存在する．このことから，本研究では Tree-Transformer の教師なし学習に加え，構文解析結果の教師データを Transformer の各階層での出力と構文解析の教師データとの損失を利用する階層的誤差逆伝播法を提案し，Transformer を用いた統語構造解析の教師あり学習手法を開発する．

2　Tree-Transformer

Tree-Transformer [3] は従来の Transformer と異なり，全ての単語に対する網羅的な self-attention 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” モジュールを導入している．通常の self-attention とは異なり，各単語が同じ consituent と呼ばれる構成要素に属する単語同士のみで self-attention を作用させる（図 1 左）．上層に移るにつれ constituent は隣同士でマージされていき，最上層では全ての単語が同じ consituent に属し，単語の依存関係すべてを捉えた状態を表現する．“Constituent Attention” モジュールでは，各単語に対して consituent の区切りを表す breakpoint を推測する確率 $a = \{a_1, ..., a_i, ...\}$ を生成する．それによって，単語 i と単語 j が同じ構成要素に含まれるかを推定する “Constituent Priors” が式 1 で示される確率をもって生成される（図 1 右）．

$$C_{i,j} = \prod_{k=i}^{j-1} a_k \tag{1}$$

式 1 に基づき，隣接する単語同士の self-attention が式 2 のように求められる．

$$E = C \odot softmax\left(\frac{QK^T}{d}\right) \tag{2}$$

学習された Tree-Transformer を用いて解析を行うことにより，以下のような統語構造の情報を含んだ出力を得ることができる．

```
(((the)(cute dog)))((is wagging))(its tail))))
```

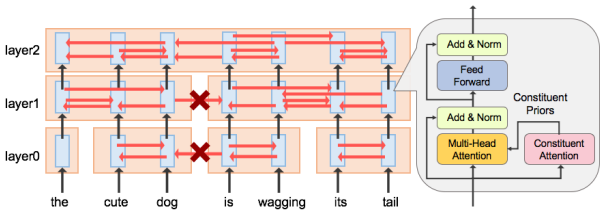

図 1:（左）3 層の Tree Transformer．ブロックは入力文から生成された constituent を表している．constituent のサイズは層ごとに徐々に大きくなる．（右）Tree Transformer の各エンコーダの構成．

3　提案手法

3.1　概要

本研究では，Tree-Transformer での Masked Language Modeling（MLM）による教師なし学習に加えて，統語構造解析結果のデータを用いる教師あり学習を行う．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出に基づき，出力を修正するための誤差逆伝搬によってモデルの結合荷重を調整しているのに対し，本研究では中間層において教師データとの損失を捉える，階層的誤差逆伝搬を提案する．

3.2　構文解析情報

教師データとして S 式で表現された統語構造データを NLTK[1] の解析モジュールを用いて再帰的に解析したものを用いる．構文木の各層の各単語 w_i の確率 a_i に対し，w_i と w_{i+1} が異なる constituent に属する，すなわち a_i が breakpoint であれば 0 を，w_i と w_{i+1} が同じ constituent に属する場合は 1 として表す．出力例を図 2 に示す．

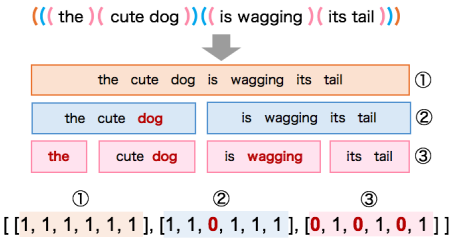

図 2: 出力例

3.3　階層的誤差逆伝播

特定の層数のエンコーダを作成し，層ごとに与えられる教師データに合わせて対象とする層を決定し，階層的に誤差逆伝播を行う．作成した教師データと各層の単語間分割確率である a との差分を捉え，損失を計算する．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．階層的誤差逆伝播のアルゴリズムを Algorithm 1 に示す．

[1] https://www.nltk.org/

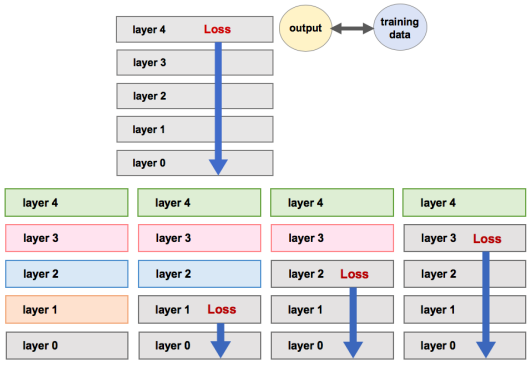

図 3: （上）通常の誤差逆伝播（下）5 層のエンコーダを作成し、教師データが 4 層の時の階層的誤差逆伝播

Algorithm 1 Supervised Parsing with Hierarchical Error Back Propagation

$L \leftarrow$ training data layer size
$model1 \leftarrow$ whole model
$model2 \leftarrow$ training data layer size model
$a \leftarrow$ link probabilities
$teach \leftarrow$ training data a
$target \leftarrow$ masked word answer
$\alpha \leftarrow$ hyper parameter
for $l = 0 \ldots L-1$ **do**
    $model2$.layers = $model1$.layers[:1+l]
    _, a=$model2$.forward
    loss_a = LOG-MSE Loss(a, $teach$)
    if l=$L-1$ **then**
        out, _=$model2$.forward ▷ MLM output
        loss_m = CrossEntropy Loss(out, $target$)
        loss = α × loss_m + (1-α) × loss_a
        loss.backward
    else
        loss_a.backward

4 実験

4.1 実験設定

構文解析データとして，構文木が 3〜5 階層で構成された，1 文につき 3〜6 単語からなる 480 文を生成した人工データを用いる．人工データは文脈自由文法（CFG）を用いて文生成を行い，構文解析情報として生成された文を Stanford Parser[2]を用いて解析を行った結果を用いた．この際，解析結果に誤りがないことを確認し，正解データとして採用した．また，階層的誤差逆伝播を行う際，文によって教師データの層数は変わるため，同じ層数が固まるようデータを予めソートした．a における損失には平均二乗誤差の対数をとり，MLM にはクロスエントロピー損失をそれぞれ用いる．また，Constituent Attention と Transformer の隠れ層を 16 次元，Multi-Head Attention の数を 4，feed-forward 層を 64 次元として 5 分割交差検証を行う．その他ハイパーパラメータの設定を表 1 に示す．

4.2 実験結果

MLM を用いた教師なし学習と a に対する教師あり学習の両方の損失を線形和を用いて束ねる際のハイパーパラメータを α とする．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．構文木の深さが異なるものは除き，その中で total_{acc} は全ての a が正解データと完全一致した割合（最終出力として正しい構文解析ができた割合），また total_a_{acc} は全層において a ごとで一致した割合を示す．5 分割交差検証の内，最高値と最低値は除いた 3 つの値で平均をとった結果を表 2 に示す．

表 1: 実験設定

交差検証	
学習量	10,000step
バッチサイズ	32
ネットワーク層数	5
勾配法	Adam
学習率	0.001
eps	1e-8
勾配閾値	1.5

表 2: 実験結果

Model (α) (α × loss_m+(1-α)× loss_a)	total_{acc} [%]	total_a_{acc} [%]
0.0	12.5	72.01
0.2	12.5	70.46
0.3	15.27	73.68
0.4	11.45	65.47
0.5	12.15	70.11
1.0(Tree-Transformer)	25.0	30.46

4.3 考察

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．しかし，Tree-Transformer と比べて total_a_{acc} で高い値を出していることから，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．また，Tree-Transformer の結果を見ると total_{acc} で高い値を出していることから MaskedLM が構文解析において機能しており，単語トークンが構文決定に大きな役割を担っていると言える．

5 まとめ

本研究では，階層的誤差逆伝播を提案し，統語構造の中間層を利用した教師あり学習手法を開発した．実験の結果，Transformer を使った統語構造解析には単語に対する self-attention 機構の役割も重要であることが確認され，MaskedLM を使った教師なし学習と構文解析情報の教師あり学習において，それぞれの損失の値域を考慮しながら重み付き和のハイパーパラメータを調査する必要があると考える．今後は教師なし/あり学習それぞれの損失の捉え方の再検討および実データを用いた統語解析を行い，提案手法の改良を進める．

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of the 31st NIPS*, 2017.

[2] Jacob Devlin et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT2019*, pages 4171–4186, 2019.

[3] Yau-Shian Wang, Hung yi Lee, and Yun-Nung (Vivian) Chen. Tree transformer: Integrating tree structures into self-attention. In *EMNLP*, 2019.

[2]https://nlp.stanford.edu/software/lex-parser.shtml