Vol.2021-NL-251 No.20
Vol.2021-SLP-139 No.20
2021/12/2

# 簡易小型化BERTを利用した日本語構文解析

河野 慎司[1,a)]　新納 浩幸[1,b)]

**概要**：BERT は fine-tuning することで様々な NLP タスクに対して高精度を出せる事前学習済みモデルであるが，fine-tuning には多くのパラメータを調整する必要があるため学習や推論に時間がかかるという問題がある．本論文では日本語構文解析に対して，BERT の一部の層を削除した簡易小型化 BERT の利用を提案する．実験では，京都大学ウェブ文書リードコーパスと京都大学テキストコーパスを混合したデータを用いて，京大版の BERT とそこから構築した簡易小型化 BERT の解析精度と処理時間を比較した．提案する簡易小型化 BERT では，京大版の BERT からの精度劣化をウェブコーパスで 0.2%，テキストコーパスで 0.79%に押さえながら，学習時間は 82%，推論時間はウェブコーパスで 66%，テキストコーパスで 84%まで削減することができた．

# Japanese parsing using simplified miniaturized BERT

Kono Shinji[1,a)]　Shinnou Hiroyuki[1,b)]

## 1. はじめに

自然言語処理における構文解析とは単語間の係り受け関係を明らかにするタスクのことである．構文解析は KNP などの日本語構文解析ツールを使用し，解析することが可能であるが，近年，BERT を構文解析に使用したとき，KNP よりも高い精度を出すことが分かってきている [1]．

BERT[2] は fine-tuning することで様々な NLP タスクに対して高精度を出せる事前学習済みモデルである．BERT を利用した構文解析では，BERT からの出力ベクトルを順伝播型ニューラルネットワーク (FFNN) に入力し構文解析を行う．ただし，fine-tuning には多くのパラメータを調整する必要があるため学習や推論に時間がかかるという問題がある．

そこで本研究では BERT の一部の層を削除した簡易小型化 BERT の利用を提案する．実験では，京都大学ウェブ文書リードコーパスと京都大学テキストコーパスを混合したデータを用いて，京大版の BERT とそれを用いて構築した簡易小型化 BERT の解析精度と処理時間を比較した．提案する簡易小型化 BERT では，3〜10 層目を削除した合計 4 層のモデルが，京大版の BERT からの精度劣化をウェブコーパスで 0.2%，テキストコーパスで 0.79%に押さえる結果となり，層を削除した後でも高い精度を維持していることが分かった．また学習・推論時間は削除する層を増やすほど速くなり，合計 4 層モデルでは学習時間は 82%，推論時間はウェブコーパスで 66%，テキストコーパスで 84%まで削減することができた．

## 2. 関連研究

BERT を使った構文解析の関連研究として柴田ら [1] の研究がある．柴田らは事前学習済み BERT を京都大学テキストコーパスと京都大学ウェブ文章リードコーパスという 2 種類のコーパスで fine-tuning し，BERT による構文解析の精度を調査した．結果，BERT を利用した場合，KNP や CaboCha，BiLSTM を利用したときよりも高い精度を出している．宇田川ら [3] は，BERT からの出力ベクトルに加え，係り元と係り先の距離などを one-hot ベクトルで表現した基本素性を FFNN に入力し，構文解析を行っている．このとき BERT と基本素性を組み合わせた場合，CaboCha よりも高い精度となっている．

[1] 茨城大学大学院理工学研究科情報工学専攻
Ibaraki University, Nakanarusawa 4-12-1, Hitachi, Ibaraki 3168511, Japan
[a)] 20nm709n@vc.ibaraki.ac.jp
[b)] hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

Hassan ら [4] は，事前学習済み BERT の層を削除し，fine-tuning した場合，GLUE タスクにおいて 12 層の BERT に比べどの程度差が出るのかを調査した．このとき，

- 4 層減らすとき，下位の層を削除するのが一番スコアが低いが，他はあまり差がない．
- 6 層減らすとき，上位の層を削除するのが一番スコアがいい．

など，削除する層数や位置によってスコアに差が出ることを述べている．

BERT の層がどの情報を捉えているかについては，Hassan らは上位層はタスクに特化した重みになっており，中間層は重要な情報を含んでおらず，下位層は文脈情報を含んでいると述べている．同じく tenney ら [5] も，基本的な構文情報は BERT の早い段階で処理され，高レベルの意味的情報は より高い層に現れると述べている．一方で，jawahar ら [6] は Probing タスクにおいて，BERT が統語的な情報を中間層が捉えているという結果を出している．hewitt ら [7] も依存構造解析において $\text{BERT}_{\text{BASE}}$ は 6~9 層，$\text{BERT}_{\text{LARGE}}$ は 15~17 層等，比較的中間に近い層が構文情報を捉えていると述べている．

## 3. BERT

本研究で使用する BERT は，京都大学 黒橋研究室が公開している BERT 日本語 Pretrained モデルの BASE 通常版を使用する．このモデルは日本語テキストのみの Wikipedia を subword に分割し，事前学習されたモデルである．

### 3.1 構文解析

本研究で BERT による構文解析を行うモデル図を図 1 に示す．まず，BERT に基本句区切りにしたトークンを subword に分割し入力する．次に注目したトークンより，後ろのトークン全てに対して *score* を計算し，*score* を元に係り受けの有無を 2 値分類する．*score* の計算には，宇田川ら [3] の構文解析モデルを元に作成した．

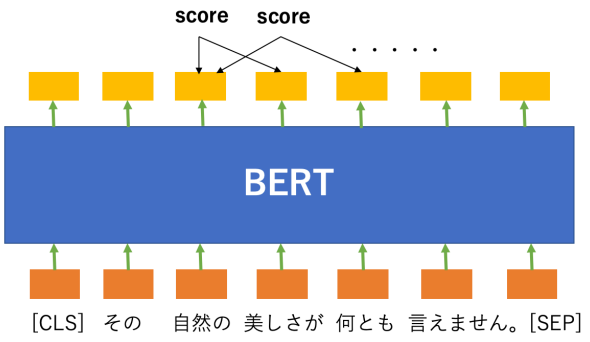


図 1: BERT による構文解析のモデル図

具体的に「私は茨城大学の学生です.」という文を本手法で構文解析した手順を述べる．まずこの文を基本句区切りにしたときの係り受け関係は図 2 となる．


図 2: 係り元と係り先の関係

本手法では，注目したトークンより後ろのトークン全てに対して係り受けの 2 値分類を行うため，注目トークンとそれ以降のトークンに対して，係り受けラベルを付与する．表 1 では，「私は」のトークンに注目したときの係り受けラベルの付与について示している．このとき subword に分割されたトークンに対しては係り受けラベルは付与しない．

表 1: 係り先ラベルの設定

| 係り元トークン | 係り先トークン | 係り受けラベル |
|---|---|---|
| 私は | 茨城 | 0 |
| 私は | 大学の | 0 |
| 私は | 学生です | 1 |

次に *score* の計算方法について述べる．$v_i$ と $v_j$ をそれぞれ係り元，係り先の BERT からの出力ベクトルだとすると，以下の式で score を出力する．このとき，$U$ と $w$ は fine-tuning する際，新しく追加したパラメータである．

$$v_{ij} = concat(v_i, v_j)$$
$$x = sigmoid(Uv_{ij})$$
$$score = wx$$

最後に，*score* を 2 値クロスエントロピーとして，係り受けの有無を fine-tuning で学習させる．

### 3.2 層の削除


図 3: BERT の位置関係

構文解析の精度と学習・推論速度の差を調査するため，京大版 BERT の一部層を削除したモデルを複数作成する．

図 3 に $\text{BERT}_{\text{BASE}}$ の位置関係を示す．BERT の入力に近い方を下位，出力に近い方を上位と呼ぶこととする．

図 4 には実験で使用するモデルの種類を表している．図 4 の灰色の層が削除した層を表しており，左から上位，中位，下位の位置を削除したものである．削除する層数はそれぞれの位置で 4，6，8 層とした．

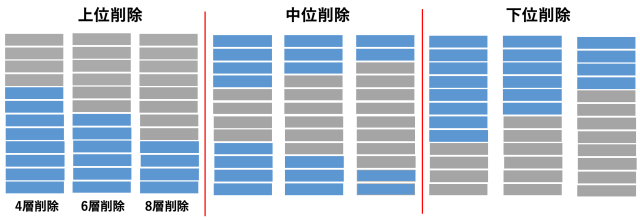


図 4: 削除する層の位置

## 4. 実験

本実験では，京都大学ウェブ文書リードコーパス (以降，Web コーパス) と京都大学テキストコーパス (以降，新聞コーパス) を混合したデータを用いて fine-tuning を行い，その後，それぞれのコーパスでテストをし，構文解析の精度を調査した．また fine-tuning 時には EarlyStopping を採用した．検証データの精度が最高値を更新しなくなってから 5 エポック後に停止させる．

### 4.1 使用データ

実験で使用するデータ数を表 5 に示す．学習と検証は Web コーパスと新聞コーパスを混合したものを利用するが，テスト時はそれぞれコーパスを分けてテストを行った．

ただし，新聞コーパスは Web コーパスに比べ，1 文のトークン数が多いものが多かったため fine-tuning では学習が収束しなかった．そのため，fine-tuning に用いる新聞コーパスのうち，1 文のトークン数が 24 個以下のもののみを使用した．テスト時のデータはトークン数で制限をかけていない．

表 2: 使用したデータ数

| | | |
|---|---|---|
| 学習 | Web・新聞 混合 | 10,000 件 |
| 検証 | Web・新聞 混合 | 2,000 件 |
| テスト | Web・新聞 別々 | 3,000 件 |

### 4.2 実験結果

各モデルの 2 つのコーパスでのテスト結果を図 5 に示す．通常 12 層の $\text{BERT}_{\text{BASE}}$ では Web コーパスで **94.00%**，新聞コーパスで **95.56%**となった．図 5 より，削除する層数を増やした場合でも，$\text{BERT}_{\text{BASE}}$ に比べ大きく精度が落ちていないことが分かった．また，削除する層数に関わらず，中位を削除したモデル (上位と下位を使用したとき) が高い精度を維持していることが分かる．

3~10 層目を削除した合計 4 層のモデルが，Web コーパスで 93.80%，新聞コーパスで 94.77%であり，$\text{BERT}_{\text{BASE}}$ からの精度劣化を Web コーパスで 0.2%，新聞コーパスで 0.79%に押さえる結果となった．

簡易化した BERT との比較だけでなく，KNP と word2vec との比較結果を図 6 に示す．図 6 の「下位 4 層削除」は先程の簡易化した BERT で Web コーパスの精度が一番低かったもの，「上位 4 層削除」は簡易化した BERT で新聞コーパスの精度が一番低かったものである．

KNP に比べ，$\text{BERT}_{\text{BASE}}$ の方が Web コーパスで約 5%，新聞コーパスで約 3%精度が良い結果となった．BERT 使った構文解析では，$v_i$ と $v_j$ を BERT からの出力ベクトルとして構文解析を行ったが，この $v_i$ と $v_j$ を BERT からの出力ベクトルではなく，word2vec で得たベクトルを利用した場合，$\text{BERT}_{\text{BASE}}$ よりも精度がかなり落ちていることが分かる．

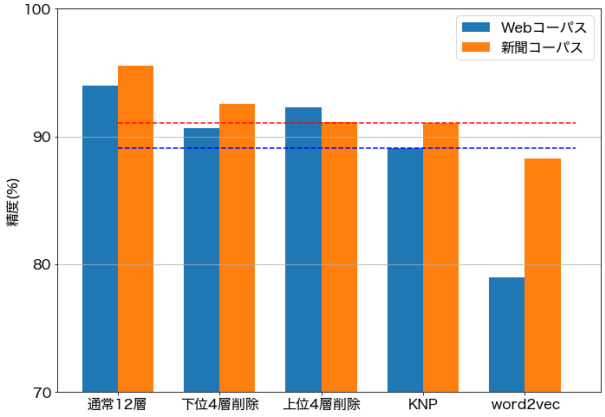


図 6: word2vec，KNP との比較結果

次に学習・推論時間の結果について述べる．学習時間の結果は表 3 の通りである．EarlyStopping を採用したため，1 エポックの平均学習時間となっている．結果より削除する層数が増えるほど，学習時間は短くなっている．

推論時間の結果は表 4 の通りである．学習時間と同じく，推論時間に関しても削除する層数が増えるほど，推論時間は短くなっていることが分かる．この時，新聞コーパスが Web コーパスよりも推論時間が長いのは，新聞コーパスの方が 1 文の平均トークン数が多いためである．

表 3: 学習時間

| | 1ep の平均学習時間 | 12 層との比較 |
|---|---|---|
| $\text{BERT}_{\text{BASE}}$ | 3 分 59 秒 | - |
| 4 層削除 | 3 分 35 秒 | -24 秒 |
| 6 層削除 | 3 分 27 秒 | -32 秒 |
| 8 層削除 | 3 分 16 秒 | -43 秒 |

8 層削除の合計 4 層モデルでは学習時間は 82%，推論時間は Web コーパスで 66%，新聞コーパスで 84%まで削減することができた．

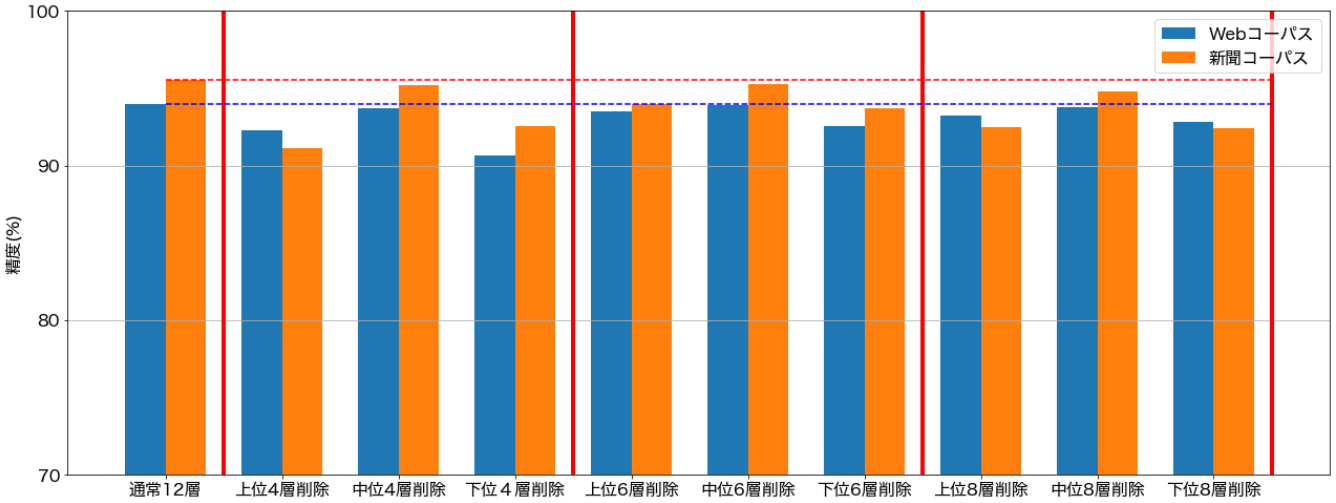


図 5: 各モデルのテスト結果

表 4: 推論時間

| | 推論時間 (Web/新聞) | 12 層との比較 |
|---|---|---|
| $\text{BERT}_{\text{BASE}}$ | 36 秒 / 79 秒 | - |
| 4 層削除 | 30 秒 / 73 秒 | -6 秒 / -6 秒 |
| 6 層削除 | 27 秒 / 71 秒 | -9 秒 / -8 秒 |
| 8 層削除 | 24 秒 / 67 秒 | -12 秒 / -12 秒 |

## 5. 考察

図 5 の各モデルのテスト結果より，上位層と下位層を利用したとき，構文解析において高い精度を維持しているため，上位層と下位層が構文情報を捉えていると考えられる．

関連研究では BERT において，Hassan ら [4]，tenney ら [5] は構文情報は下位層を含んでいると述べているが，jawahar ら [6]，hewitt ら [7] は中間層が構文情報を捉えていると述べている．

本研究において，BERT のどの層が構文情報を捉えているのか，より細かく調査するため，$\text{BERT}_{\text{BASE}}$ の中で 11 層を削除し，1 層目から 12 層目まで 1 層ずつ fine-tuning し実験を行った．その結果，新聞コーパスのテスト結果を図 7，Web コーパスでのテスト結果を図 8 に示す．新聞コーパスは下位の層，上位の層が，5 層から 8 層目の中間の層よりも精度がいいことがわかった．一方で Web コーパスに関してはどの位置の層を同程度の精度であった．

今回利用したコーパスの平均トークン数と 1 文の平均未知語数は表 5 に示す．新聞コーパスの方が Web コーパスに比べ，1 文の平均トークン数が長く，それに伴い平均未知語数を多くなっている．このことから，コーパスの種類や，1 文のトークン数と未知語数が多いことが，BERT が構文情報を捉える位置に影響を与えているのではないかと考えられる．

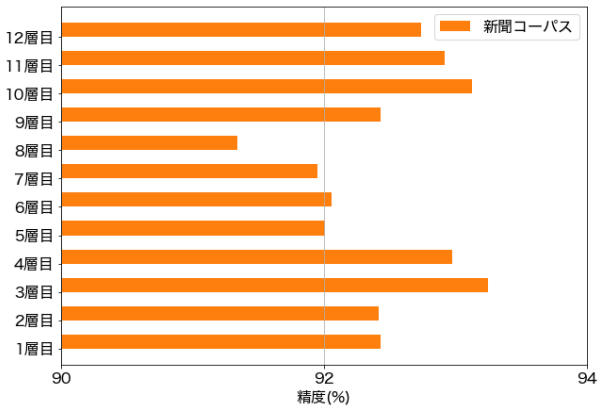


図 7: 新聞コーパスを 1 層ずつテスト

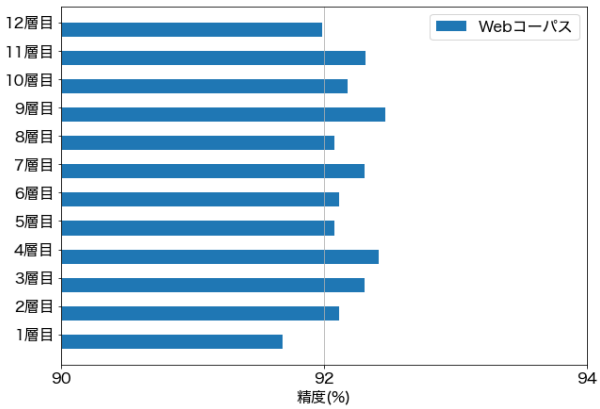


図 8: Web コーパスを 1 層ずつテスト

表 5: 各テストコーパスの特徴

| | Web コーパス | 新聞コーパス |
|---|---|---|
| 平均トークン数 | 14.85 | 23.55 |
| 1 文の平均未知語数 | 1.7 | 2.7 |

## 6. おわりに

本研究では BERT の一部層を削除し，簡易化した BERT

を用いることによる，日本語構文解析の精度の差を調べた．結果，$\mathrm{BERT}_{\mathrm{BASE}}$ が両コーパスともに精度が最も高いが，削除する層数が増えても，精度にあまり差がでなかった．このとき上位層と下位層を利用するモデルが高い精度を維持していることがわかった．また，削除する層数を増やすほど，学習・推論にかかる時間は短くなった．

今後は各テストコーパスの平均トークン数がほぼ同じになる文でテストを行い，BERT のどの層が構文情報を捉えているのかを調べていく事と，$\mathrm{BERT}_{\mathrm{LARGE}}$ においても精度の違いを調査していきたい．

## 謝辞

本研究は JSPS 科研費 JP19K12093 および 2021 年度国立情報学研究所公募型共同研究（2021-FC05）の助成を受けています．

### 参考文献

[1] 柴田知秀, 河原大輔, and 黒橋禎夫. BERT による日本語構文解析の精度向上. **言語処理学会 第 *25* 回年次大会 発表論文集**, pages 205–208, 3 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] 宇田川忠朋, 久保大亮, and 松崎拓也. BERT を用いた日本語係り受け解析の精度向上要因の分析. **人工知能学会第 *35* 回全国大会論文集**, 6 2021.

[4] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models, 2021.

[5] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline, 2019.

[6] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics.

[7] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.