深層学習を用いた階層的テキストセグメンテーションモデル

久島務嗣 [1] 吉田尚水 [1] 小林優佳 [1]
永江尚義 [1] 岩田憲治 [1]
[1] 株式会社　東芝　研究開発センター
tsuyoshi2.kushima@toshiba.co.jp

概要

近年では，大規模コーパスと深層学習を用いた高性能なテキストセグメンテーションモデルが提案されている．既存モデルは 1 つの階層でのみセグメンテーションを行うが，文章は章立てや段落といった階層構造を持つ．この構造情報を活用することで，推定時に考慮できる情報が増え，モデルのドメイン汎化性能が向上すると期待される．そこで，構造情報を持った大規模コーパスを構築し，深層学習を用いた階層的テキストセグメンテーションモデルを提案する．学習コーパスとドメインの異なるコーパスで評価した結果，提案モデルは既存モデルの性能を上回り，構造情報の有効性を確認した．

1　はじめに

書籍や文書といった人によって作られた文章は，章立てや段落などの構造を持つ．この構造情報は，高度なテキスト分析において有用である．しかし，音声認識の様な自動化技術の発達によって構造を持たないテキストが大量に存在するようになった．構造を持たないテキストの分析は容易でないため，高性能なテキスト構造化技術が求められている．

テキスト構造化技術の 1 つにテキストセグメンテーション (Text Segmentation: TS) がある．これは，テキストを意味的に関連性がある複数の文のまとまり (セグメント) に分割し，構造化する技術である．TS には大きく 2 つのタイプがある．1 つは，テキストからセグメント終端となる文を検出し，テキストをセグメント系列に変換する線形的 TS(Linear TS: LTS, 図 1 下) である．もう一つは階層的な構造化を行う階層的 TS(Hierarchical TS: HTS, 図 1 上) である．古典的な LTS・HTS モデルは文の類似度や単語の一貫性に基づいた教師なし学習モデルで，人工的に生成したデータでは高い性能を発揮していたが，自然なデータでは性能が低いことが課題であった [1, 2, 3, 4].

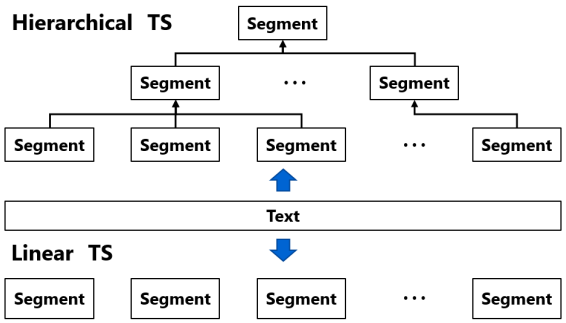

図 1　LTS と HTS の概要

近年では，大規模コーパスを用いた深層学習 LTS モデルがいくつか提案されており [5, 6, 7, 8], 自然なデータでも高性能なテキスト構造化が可能となってきた．一般に文章の持つ構造は階層構造であるため，セグメントに含まれる文の数 (大きさ) は構造化される階層によって異なる．例えば，節の下にいくつかの段落がある場合，下の階層で構造化すれば 1 つのセグメントが段落になり，上の階層で構造化すれば 1 つのセグメントが節になる．適切なセグメントの大きさは構造化後の分析に依って異なるため，汎用的に利用するには階層的な構造化が必要となる．しかし，LTS モデルは階層的な構造化が出来ず，対応した階層別にモデルを用意するのはコストが高い．

深層学習を用いた HTS モデルが提案されていない原因として，構造情報を持った十分大きい学習コーパスが構築されていないことが挙げられる．大規模コーパスを用いて HTS モデルを深層学習することで，様々な分析に汎用的に利用できる高性能な HTS モデルが学習できる．さらに，テキストが持つ階層構造を活用することで，推定時に考慮できる情報が増え，LTS モデルよりもテキストのドメインに対する汎化性能が向上することも期待される．そこで，章・節・段落の 3 つの階層の構造情報を持った大規模な学習コーパスを構築し，深層学習を用いた 3 層の HTS モデルを提案する．

まず，2 章で関連研究を挙げ，3 章で構築したコー

パスと提案モデルについて説明する．4 章では実験設定と評価方法, 5 章では結果と考察, 6 章では最後に本稿のまとめと今後の課題について述べる．

2 関連手法

深層学習を用いた LTS モデルは文特徴ベクトル生成部 (encoder) とセグメント終端推定部 (predictor) から成る．いずれの部分も入力は系列となるので, LSTM や Transformer Encoder(TFE) が用いられる．TFE を用いた LTS モデルには Glavăs ら [8] と Lukasik ら [7] のモデルがある．Glavăs らは, 再帰的なモデル (LSTM など) よりも TFE の方が自然言語タスク全般で性能が良く収束が早いために, TFE をモデルに採用した．また, Lukasik らは encoder・predictor の両方に学習済み BERT を用いた HierBERT を提案した．

HTS モデルには Kazantseva ら [3] と Bayomi ら [4] のモデルがある．いずれのモデルも文の類似度に基いた階層クラスタリング手法を用いた教師なし学習モデルで, 小さいセグメントから推定をはじめ, その結果を用いてだんだん大きいセグメントを推定する, ボトムアップ方式の構造化を行った．

3 提案手法

3.1 コーパス

深層学習を用いて HTS モデルを学習するためには , 2 つ以上の階層を持つ大規模なコーパスが必要となる．しかし, その様なコーパスは存在しないため, Section[9] と Wiki-40B[10] を用いて新たなコーパスを構築した．Section は LTS の学習に広く用いられているコーパスで, Wikipedia の都市と疾病に関する記事を収集することによって構築された．記事は SPARQL を使用して対応するトピックの WikiData の QID を検索する事によって収集された．また, Wiki-40B は Wikipedia を前処理したコーパスで, 節と段落の 2 つの構造情報と WikiData の QID が付与されている．Section と同じ記事のデータを Wiki-40B から抽出することで, 構造情報を持った Section(Hier.Section) を構築した．

さらに, Wiki-40B は 1 つのデータに複数の節を含むため, 1 つのデータを章とみなすことができる．そこで, 1 つのデータのテキスト全体を章のセグメントとし, その終端に章のセグメント終端の正解ラベルを付与した. したがって, Hier.Section は章・節・段落の 3 つの構造情報を持つ．

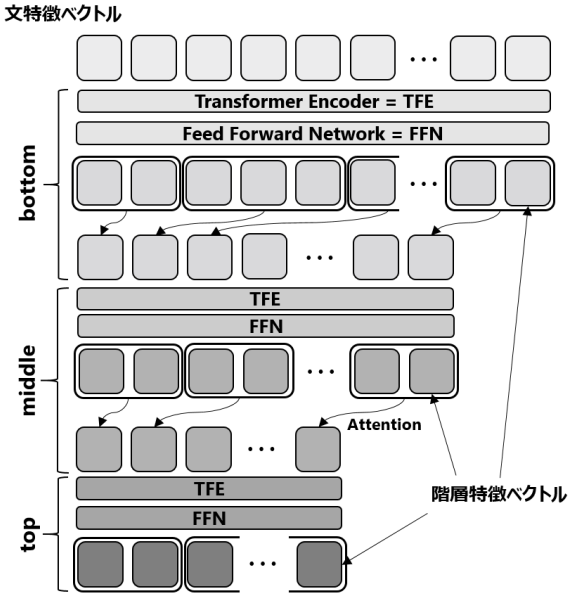

図 2 Predictor モデル概要図

データの抽出は, Section の各データに付与された WikiData の URL から特定した QID を用いて行った．抽出したデータの前処理として, 複数の文が連結されていた場合には分割し, 節のセグメント数が 2 つ以下のデータと 3 文以下のデータはコーパスから削除した．文の分割には Python のオープンソースライブラリ nltk[1)] を使用した. 学習・検証・評価データへの分割は Section の学習・検証・評価データと同じデータが含まれるように行った．

また, LTS の評価に良く用いられるコーパスに Elements[11] がある．Elements は複数の文が連結されているデータが多く, また, 記号が削除されているために元の文に分解することが難しいことから, Wiki-40B から新たに構築した Hier.Elements を評価データとして使用した．Hier.Elements は化学元素に関する記事の QID を SPARQL を使用して収集し, 対応するデータを Wiki-40B から抽出することによって構築した. 前処理は Hier.Section と同様に行った．

3.2 提案モデル

HTS では階層ごとに関連性を考慮する範囲が異なるため, 局所的・大域的な関連性がどちらも重要である．TFE は系列全体の関連性を見ながら特徴を抽出するため, HTS モデルの predictor には TFE が適切である．さらに, encoder に事前学習済み BERT を用いることで, predictor では学習コーパスに依存しない汎用的な文特徴ベクトルが利用できる．また, 小さ

1) https://www.nltk.org/

いセグメントはテキストを構成する意味の最小単位となるため, ボトムアップで構造化することで, その後のセグメントの統合を精度よく行うことができる [4]. したがって, 本稿は encoder には BERT, predictor には TFE を用い, ボトムアップで推定する HTS モデルを提案する.

encoder は BERT が出力した sub-word 単位の特徴ベクトルを平均し, 文特徴ベクトル f^i として出力する. predictor のモデル概要図を図 2 に示す. predictor は 2 層の TFE を 3 つの持ち, それぞれのパラメータは独立している. bottom・middle・top は段落・節・章の階層にそれぞれ対応し, ボトムアップに bottom から順にセグメント終端を推定する.

bottom では, まず, encoder から受取った文特徴ベクトル系列 $f_{1:L}$ を TFE に入力し, 階層特徴ベクトル系列 $h^b_{1:L}$ を得る (式 1). 計算した $h^b_{1:L}$ から Feed Forward Network(FFN) と softmax 関数によって bottom のセグメント終端確率 $p^b_{1:L}$ を算出する (式 2).

$$h^b_{1:L} = TFE^b(f_{1:L}) \in \mathbb{R}^{L\times d} \tag{1}$$

$$p^b_i = \frac{exp(g^b_i)}{\sum_{j=1}^{L} exp(g^b_j)} \in \mathbb{R}^2 \tag{2}$$

$$g^b_{1:L} = FFN^b_p(h^b_{1:L})$$

式中の d は BERT の特徴ベクトル次元, L は入力テキストの文の数をそれぞれ表し, $FFN^b_p(\cdot) = W^b_p \cdot relu(\cdot) + b^b_p$ である.

次に, $p^b_{1:L}$ を閾値処理しセグメント終端位置を特定する. ただし, 学習時には正解ラベルをセグメント終端位置として用いた. 特定したセグメント終端位置からセグメント系列 Seg^b を得る. セグメント Seg^b_i は i 番目のセグメントに含まれる文の添え字集合を表す. 例えば, 3 文目と 6 文目がセグメント終端の場合, $Seg^b_1 = \{1,2,3\}, Seg^b_2 = \{4,5,6\}$ となり, $|Seg^b_1| = |Seg^b_2| = 3$ である. 統合ベクトル e^b_i は, コンテキストベクトル C^b を用いた注意機構 [12] によって計算する (式 3).

$$e^b_i = \frac{1}{|Seg^b_i|} \sum_{j\in Seg^b_i} a^b_j \cdot h^b_j \tag{3}$$

$$a^b_i = \frac{exp(z^b_i)}{\sum_{k\in Seg^b_j} exp(z^b_k)} \quad s.t.\ i \in Seg^b_j$$

$$z^b_{1:L} = C^b \cdot tanh(FFN^b_z(h^b_{1:L})) \in \mathbb{R}^L$$

bottom で特定したセグメント系列が $|Seg^b| = M$ であるとき, middle では bottom の計算した統合ベクトル系列 $e^b_{1:M}$ を入力として受け取る. 同様に, middle で特定したセグメント系列が $|Seg^m| = N$ であるとき, top は middle の統合ベクトル系列 $e^m_{1:N}$ を受け取る. 入力ベクトル以外は bottom と同様の手続きで middle・top のセグメント終端確率を推定する. 目的関数は各階層における Cross-Entropy Loss(CEL) の総和としてモデルを更新した.

$$loss = \frac{1}{L}CEL(y^b_{1:L}, p^b_{1:L}) + \frac{1}{M}CEL(y^m_{1:M}, p^m_{1:M}) + \frac{1}{N}CEL(y^t_{1:N}, p^t_{1:N}) \tag{4}$$

式中の y はそれぞれの階層の正解ラベル,$p^m_{1:N}$・$p^t_{1:N}$ は middle・top で計算されたセグメント終端確率をそれぞれ表す. 提案モデルでは各階層のセグメント単位で特徴ベクトルを統合しながら推定を進めるため, 階層ごとに系列長が異なる. そのため, 階層ごとのロスのバランスを取るために系列長でスケーリングを行った. また, FFN に入力される特徴ベクトルは, 入力される直前に一定の割合で要素を 0 に置換する dropout を適用した [13].

4 実験

4.1 実験設定

実験に使用するコーパスとその統計量を表 1 に示す. 学習コーパスとトピックが異なるコーパスをドメイン外 (Out-Of-Domain: OOD†) とする. 学習コーパスには Hier.Section を用いた. 学習・評価データの作成の詳細は付録を参照されたい. 最適化関数には AdamW を用い, 学習率は 10e-4, 学習エポック数は 50, dropout 割合は 0.1 とした. 評価時のセグメント終端位置の特定に使用する閾値は, 検証データにおいて最も良い F1 となった閾値を使用した. 実装には Pytorch[2] と hugingface[3] の Transformers, encoder の BERT には"bert-base-cased" を使用した. また, BERT はパラメータを固定して実験を行った.

ベースラインは, LSTM を積層した HierLSTM[5] と, BERT を積層した HierBERT[7] の 2 つの LTS モデルとした. HierLSTM と HierBERT はそれぞれの論文にしたがって実装し, 提案モデルと同様の学習・評価を行った. また, HierBERT の BERT には提案モデルと同様に"bert-base-cased" を使用した.

2) https://pytorch.org/
3) https://huggingface.co/docs/transformers/index

表 1　コーパス統計量

	学習コーパス	評価コーパス			
コーパス	Hier.Section	Cities[11]	Hier.Elements†	Wiki-50†[5]	Choi†[14]
トピック	都市・疾病	都市	化学元素	-	-
データ (長さ / 数)	48.0 / 17,844	62.8 / 100	125.2 / 123	56.4 / 50	70.0 / 700
節 (長さ / 数)	7.3 / 6.5	5.1 / 12.2	10.3 / 12.0	8.3 / 6.7	6.9 / 10.0
段落 (長さ / 数)	3.1 / 15.4	- / -	3.3 / 37.4	- / -	- / -

表 2　評価コーパスの評価結果 (F1 ↑ / PK ↓)

Models	Cities	Hier.Elements†	Wiki-50†	Choi†	OOD-Ave
Ours	0.688 / 0.200	**0.479 / 0.284**	0.425 / 0.327	**0.750 / 0.172**	**0.551 / 0.261**
HierBERT	**0.717** / 0.166	0.472 / 0.312	0.471 / 0.334	0.642 / 0.238	0.529 / 0.294
HierLSTM	0.709 / **0.165**	0.389 / 0.355	**0.486 / 0.324**	0.640 / 0.237	0.505 / 0.305

4.2　評価

評価には F1・ Pk[15] の 2 つの指標を用いた．Pk は窓幅 k 内のセグメント終端数の一致を測ることでセグメントのエラー率を計算する．F1・Pk は 1 つのコーパスのすべてのデータの推定結果を連結した系列に対して計算した. Pk は Python のオープンソースライブラリ segeval[4] を用いて計算した.

評価コーパスには表 1 に示した 4 つのコーパスを用いた．複数階層の正解ラベルを持たないコーパスに対する提案モデルの評価は，節の階層の推定結果を用いて行った．また，各データの末尾は必ずセグメント終端となるため，節と段落の階層では既知のものとして評価を行った.

5　結果と考察

表 2 に評価コーパスにおける結果を示す．表中の太字は最も良いモデルの結果を表し，OOD-Ave には OOD コーパスの結果の平均を示す.

個々の OOD コーパスの結果を見ると，Hier.Elements と Choi では F1・Pk 共に提案手法が最も良い結果となった．また，Wiki-50 においては，F1 では既存モデルと差があるが，Pk ではその差は小さい．さらに，OOD-Ave の結果をみると，F1 と Pk 共に提案モデルが最も良い結果となった．したがって，構造情報を用いることでモデルのドメイン汎化性能が向上することが確認された.

表 3 に Hier.Section の評価データ (Hier.Section-test) における結果を示す．Hier.Section-test における提案モデルの結果を見ると，いずれの階層においても高い性能で推定出来ている．したがって，深層学習を用いることで学習コーパスにおいては高性能な HTS モデルを学習出来ることが確認された.

表 3　Hier.Section-test の評価結果 (F1 ↑ / PK ↓)

Models	章	節	段落
Ours	0.801 / 0.091	0.733 / 0.168	0.761 / 0.195
HierBERT	- / -	**0.780 / 0.139**	- / -
HierLSTM	- / -	0.766 / 0.147	- / -

一方で，階層ごとの性能を比較すると，章の性能が他の階層よりも高く，階層ごとの性能差が大きい．さらに,Hier.Section-test と Cities では既存モデルの性能の方が高く，提案モデルの学習コーパスと同じドメインのコーパスでの性能は低下していることが分かる．これは，推定の難易度が階層ごとに異なることを示唆している．また，提案モデルはボトムアップ方式で推定を進めるため，段落の階層の推定精度が他の階層に影響を与える．したがって，モデル全体の性能に重要な階層よりも他の階層が先に学習されたことによって，不適切な局所解に収束した可能性がある．適切な順序で階層の学習が進むような工夫をすることでモデルの性能が向上することが期待される.

以上のことから，学習コーパスにおいては高性能な HTS モデルを学習可能であり，提案モデルはドメイン汎化性能が高いことが確認されたが，更なる性能改善の余地があることがわかった.

6　おわりに

本稿では，構造情報を持った大規模な学習コーパスを構築し，深層学習を用いた階層的なテキストセグメンテーションモデル (HTS モデル) を提案した．評価の結果，学習コーパスにおいては高性能な HTS モデルが学習可能であり，提案モデルは既存モデルよりもドメイン汎化性能が高いことから，構築したコーパスと多くの構造情報を推定に用いることの有効性が確認された．学習の工夫によってモデルの性能を更に向上させることが今後の課題である.

4)　https://segeval.readthedocs.io/en/latest/

参考文献

[1] Marti A. Hearst. Text tiling: Segmenting text into multi-paragraph subtopic passages. **Computational Linguistics**, Vol. 23, No. 1, pp. 33–64, 1997.

[2] Goran Glavăs, Federico Nanni, and Simone Paolo Ponzetto. Unsupervised text segmentation using semantic relatedness graphs. In **Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics**, pp. 125–130.

[3] Anna Kazantseva and Stan Szpakowicz. Hierarchical topical segmentation with affinity propagation. In **Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics**, pp. 37–47, 2014.

[4] Mostafa Bayomi and Sěamus Lawless. C-hts: A concept-based hierarchical text segmentation approach. In **Proceedings of the Eleventh International Conference on Language Resources and Evaluation**, 2018.

[5] Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics**, pp. 469–473.

[6] Linzi Xing, Brad Hackinenz, Giuseppe Carenini, and Francesco Trebbi. Improving context modeling in neural topic segmentation. In **Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing**, pp. 626–636, 2020.

[7] Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. Text segmentation by cross segment attention. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 4707–4716, 2020.

[8] Goran Glavăs and Swapna Somasundaran. Two-level transformer and auxiliary coherence modeling for improved text segmentation. In **Proceedings of the 34th Association for the Advancement of Artificial Intelligence**, 2020.

[9] Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. Sector: A neural model for coherent topic segmentation and classification. **Transactions of the Association for Computational Linguistics**, Vol. 7, pp. 169–184, 2019.

[10] Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. Wiki-40b: Multilingual language model dataset. In **Proceedings of the Twelfth Language Resources and Evaluation Conference**, pp. 2440–2452, 2020.

[11] Harr Chen, S.R.K. Branavan, Regina Barzilay, and David R. Karger. Global models of document structure using latent permutations. In **Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics**, pp. 371–379, 2009.

[12] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 1480–1489.

[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, Vol. 15, No. 56, pp. 1929–1958, 2014.

[14] Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In **1st Meeting of the North American Chapter of the Association for Computational Linguistics**, pp. 26–33, 2000.

[15] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. **Machine Learning**, Vol. 34, No. 1, pp. 177–210, 1999.

[16] Eisenstein Jacob and Regina Barzilay. Bayesian unsupervised topic segmentation. In **Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing**, pp. 334–343, 2008.

A 付録

A.1 学習・評価データ作成

学習データはランダムにサンプリングした 2〜6 個のデータを連結することによって作成した．計算量の都合で，連結したデータの文長と文数はそれぞれで sub-word 単位で 64 単語・128 文を上限に制限した．上限を超える場合は，文は上限位置で切り捨て，データは上限位置で分割した．また，節・段落の順序は 1/2 の割合でランダムな順序にシャッフルした．

評価データは，コーパス中のすべてのデータを連結し，128 文を窓として 64 文ずつスライドしてデータの半分が重複するように分割し作成した．重複する部分は推定したセグメント終端確率を平均することで統合し，最終的な推定結果として評価を行った．評価時には，節・段落の順序のシャッフルは行わず，文字数の制限のみ学習時と同様に行った．

A.2 実験結果詳細

表 4 にはベースラインとした 2 つの既存モデルの結果，表 5 には提案モデルの結果をそれぞれ示す．段落の正解ラベルを持たないコーパスでは，節の正解ラベルを使って段落階層の推定結果を評価した (表 5 下線)．

Section-test は Section の評価コーパスで，Hier.Section-test と結果に乖離がないことを確認するために評価を行った．Clinical は医学書からいくつかの章を抽出し，節をセグメントとして構築したコーパスである [16]．形式の異なるコーパスに対する汎化性能を検証するために Clinical でも実験を行った．以下に Clinical の統計量を示す．

- データ (長さ / 数): 140.4 / 227, 節 (長さ / 数): 34.8 / 4.0, 段落 (長さ / 数): - / -

Clinical は 1 つのデータが長く節のセグメントも大きい．Clinical の 1 つのセグメントは，Hier.Section の 1 つのデータと同程度の大きさであるため，提案モデルにおいては章の階層で推定されるセグメントに相当すると考えられる．そこで，Clinical ではデータの連結は行わず，データを 1 つずつ入力し推定を行った．さらに，提案モデルの章・段落の階層の推定結果は，節の正解ラベルを使って評価を行った．

A.3 階層数を固定した HTS における課題

Clinical における提案モデルの結果をみると，章の階層での F1・Pk が最も良い結果となった．既存モデルの結果と比較すると，F1 では HierBERT が最も良いが，Pk では提案モデルが最も良い結果となっている．Clinical における提案モデルの章の階層の推定結果を分析すると，湧き出し誤りは極端に少なく，見落とし誤りが失敗のほとんどを占めており，正しく検出されたセグメント終端のほとんどはテキストの末尾の既知としたセグメント終端であった．その結果，セグメント数が少ない Clinical では Pk が高くなったことが分かった (表 1)．さらに，節の階層では高い Recall でセグメント終端が検知できていることから，章の階層ではそのほとんどが棄却されていることが分かる (表 5)．これは，Clinical のセグメントは学習コーパスにおける章と節の中間の階層にあたることを示唆している．また，Hier.Elements の章の階層においても，節の階層で推定されたセグメントの終端をほとんど棄却しており，Clinical と同様の傾向が確認された．

HTS においては，この様な対応できない階層が正解として求められる場合があるが，階層数を固定した HTS モデルでは，原理上対応が難しい．汎用的に利用可能なモデルには，この様な課題に対応することも必要となるため，対応可能なモデルの検討も今後の課題である．

表 4 実験結果詳細 既存モデル

	HierBERT				HierLSTM			
Corpus	F1 ↑	PRE ↑	REC ↑	PK ↓	F1 ↑	REC	PRE ↑	PK ↓
Hier.Section	0.780	0.715	0.857	0.139	0.766	0.702	0.843	0.147
Section-test	0.771	0.725	0.824	0.151	0.763	0.711	0.823	0.156
Cities	0.717	0.675	0.764	0.166	0.709	0.644	0.788	0.165
Hier.Elements†	0.472	0.340	0.772	0.312	0.389	0.302	0.547	0.355
Wiki-50†	0.471	0.323	0.873	0.334	0.486	0.362	0.739	0.324
Choi†	0.642	0.495	0.914	0.238	0.640	0.515	0.847	0.237
Clinical†	0.416	0.476	0.369	0.319	0.298	0.442	0.225	0.431

表 5 実験結果詳細 提案モデル

	章				節				段落			
Corpus	F1 ↑	PRE ↑	REC ↑	PK ↓	F1 ↑	PRE ↑	REC ↑	PK ↓	F1 ↑	PRE ↑	REC ↑	PK ↓
Hier.Section	0.850	0.870	0.832	0.052	0.738	0.734	0.743	0.165	0.754	0.764	0.745	0.203
Section-test	-	-	-	-	0.728	0.699	0.759	0.159	0.443	0.293	0.906	0.468
Cities	-	-	-	-	0.688	0.659	0.719	0.200	0.516	0.363	0.887	0.415
Hier.Elements†	0.085	0.333	0.049	0.442	0.479	0.531	0.436	0.284	0.590	0.554	0.631	0.324
Wiki-50†	-	-	-	-	0.425	0.547	0.347	0.327	0.421	0.301	0.698	0.397
Choi†	-	-	-	-	0.750	0.800	0.706	0.172	0.531	0.382	0.873	0.320
Clinical†	0.398	0.752	0.271	0.262	0.211	0.124	0.714	0.535	0.133	0.072	0.832	0.603