

同期的テキスト補完による並列構造の生成

寺西 裕紀

松本 裕治

理化学研究所

{hiroki.teranishi, yuji.matsumoto}@riken.jp

概要

本研究は並列構造とその範囲が明示された文を生成するアプローチを提案する。提案手法は等位接続詞と空所を表すトークンを2通りの方法で文に挿入し、言語モデルを用いてそれらの空所に同一のテキストを埋めて補完する。並列構造の範囲同定タスクにおいて、生成された並列構造を効果的に選択する枠組みを開発し、低リソース状況下の学習におけるタスクの性能向上に貢献するような並列構造の事例を提案手法が生成していることを検証する。

1 はじめに

言語モデルの事前学習は大規模なラベルなしテキストから汎用的な表現を得る手段として近年発展しているパラダイムである。BERT [1] や BART [2] をはじめとする言語モデルの fine-tuning は様々なタスクの性能を底上げしたが、人手によってラベル付けされた大量の学習データではなく言語モデルを活用して効率的な学習を行う few-shot 学習 [3, 4, 5, 6] やデータ拡張 (data-augmentation) [7, 8, 9, 10] などの試みも注目されている。しかし、これらの試みの多くは文分類などの簡易なタスクへの適用に限定されており、構文解析など構造に関するタスクへの応用については研究が進んでいない¹⁾。

本研究は、言語モデルを用いて統語的な構造を生み出す試みとして、等位接続詞によって結びつけられた要素 (並列句) から成る構造 (並列構造) を生成するアプローチを提案する。並列構造は自然言語に頻出して構文的・意味的曖昧性を引き起こすため、多くのタスクにおいて誤りの要因となる。本研究によって不自然でないような並列構造が生成できれば、並列構造の範囲同定手法のさらなる精度向上も期待できる。

本研究で提案する手法は文の一部分に着目し、そ

1) 言語モデルを用いない方法で文の構造を操作してデータ拡張を行う研究 [11, 12] は試みられている。

の箇所と並列となるようなテキストを言語モデルによって生成することで並列構造を発生させる。より具体的には、等位接続詞とマスクトークンを2通りの方法で挿入した文のペアを用意し、それらの空所に入る同一のテキストをマスク言語モデルに予測させる。また、生成された並列構造とその範囲が不適当であるような事例を取り除く方法を、並列構造の範囲同定タスクの学習と統合したフレームワークとして提案する。提案手法が並列構造と範囲の組の有用な事例を生成して効果的に取捨選択することで、低リソース状況下における並列構造の範囲同定タスクの性能向上に貢献することを実験により示す。

2 提案手法

提案手法は text-infilling タスクによって事前学習された言語モデルを用いて並列構造の生成を行う。形式的には、文 $S_{1:N} = w_1, \dots, w_N$ 内に並列構造を生じさせ、並列構造を持つ文 $S'_{1:N'} = w_1, \dots, w_{N'}$ とその並列構造の構成要素のアノテーション $\langle k, \{(i^{(1)}, j^{(1)}), \dots\} \rangle$ を得る。ここで k は等位接続詞の位置であり、スパン $(i^{(m)}, j^{(m)})$ は各並列句の範囲を表す。本研究は英語において等位接続詞 *and* によって結びつけられた二つの並列句から成る並列構造を生成の対象とする。この目的を達成するために、提案手法はまず入力文 S から参照スパン (i, j) を選択し、参照スパンと対応付けられるようなテキストを言語モデルによって生成して等位接続詞とともに挿入することで、二つの並列句 $w_{i:j}, w_{i':j'}$ と等位接続詞 w_k から成る並列構造を持つ文 S' を出力する。本手法の概要を図 1 に示す。以降の小節にて手法の詳細を説明する。

2.1 参照スパンに対応するテキストの生成

提案手法は入力文 S において等位接続詞とマスクトークンを2通りの方法で埋め込んだ文 $S'^{(1)}, S'^{(2)}$ を用いる。具体的には、 $S'^{(1)}$ では等位接続詞とマスクトークンがこの順で参照スパンの後ろに挿入さ

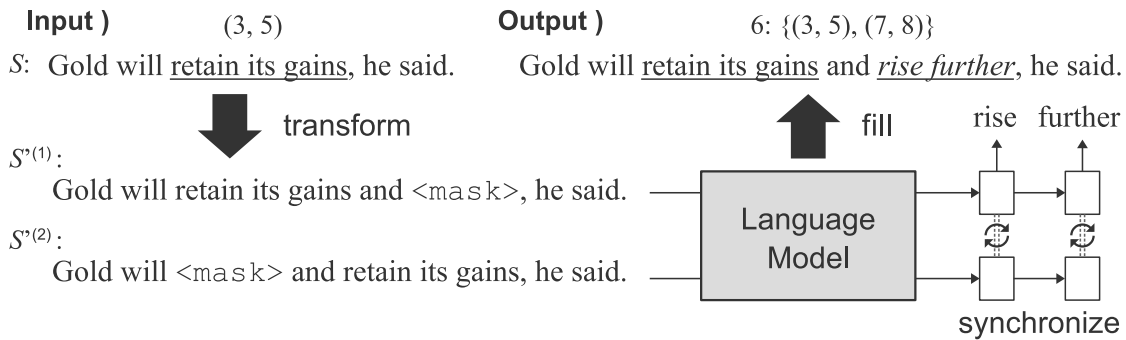


図1 提案手法による並列構造生成の概要. 入力文は指定されたスパン (“*retain its gains*”) に基づいて2通りの方法によってマスクされ, 言語モデルがそれらのマスクに対する予測を同期的に行うことで同一のテキストを当てはめる.

れ, $S^{(2)}$ では等位接続詞がマスクトークンに続く形で参照スパンの前に挿入される. この二つのマスクに同一のテキストを埋めることによって参照スパンと対応付けられたテキストを得る. この方法は並列句が類似・可換となるように [13, 14], 言語モデルが参照スパンと文脈を活用できる点で利点がある.²⁾

本研究ではマスクトークンに対するテキスト生成のために T5 [15] を用いる³⁾. T5 モデルは Transformer [16] のエンコーダ・デコーダから成り, 文のマスクされたスパンを埋めるテキストを生成するよう事前学習される. 例えば, エンコーダへの入力 “*Thank you <X> me to your party <Y> week.*” に対して, デコーダが “*<X> for inviting <Y> last <Z>*” を出力するよう訓練される. この学習方法に則って, 提案手法は文 $S^{(1)}$, $S^{(2)}$ をそれぞれ独立に T5 のエンコーダに与え, マスクトークンに当てはまるテキストをデコーダによって生成する. ただし, $S^{(1)}$, $S^{(2)}$ のマスクトークンに入るテキストが同一となるような成約下でデコーディングを行う.

$$\begin{aligned} \mathbf{u}_t^{(1)} &= \mathbf{u}_t^{(2)} = f(\mathbf{u}_t^{(1)}, \mathbf{u}_t^{(2)}) \\ T_t^{(1)} &= \arg \max(\mathbf{u}_t^{(1)}) \\ T_t^{(2)} &= \arg \max(\mathbf{u}_t^{(2)}) \end{aligned} \quad (1)$$

ここで $\mathbf{u}_t^{(1)} \in \mathbb{R}^V$, $\mathbf{u}_t^{(2)} \in \mathbb{R}^V$ (V はトークンの語彙数) はそれぞれ, 入力文 $S^{(1)}$, $S^{(2)}$ に対するデコーディングステップ t におけるトークンのスコアベクトル (logit) であり, 二つのベクトルは関数 f の働きによって同じ値を持つように同期される. $S^{(1)}$, $S^{(2)}$ のどちらか一方の文脈のみに過剰に適合するトークンが出力されることを避けるため,

- 2) $S^{(1)}$ だけを用いる方法では, 並列構造の範囲が不適当となるテキストが生じやすい (例: “*The boys and <mask> play baseball on Saturdays.*” → “[*The boys*] and [*Mary usually*] play baseball on Saturdays.”).
- 3) BERT を用いた並列構造の生成方法を Appendix A に示す.

本研究では関数 f として要素ごとに最小値をとる (element-wise min) 関数を採用する⁴⁾.

2.2 教師あり並列構造解析への適用

前節で示した方法によって並列構造を持つ文が得られるが, 並列構造の範囲が不適当な例が生じうる. そのような例は並列句間の一致を判定するような後処理によって取り除くことが望ましいが, 一致を判定するルールなどを開発するコストがかかる. 本研究では, 範囲が明示された並列構造を含む文のアプリケーションとして, 生成された事例を並列構造の範囲同定タスクの教師データとして用いる. 本節は範囲同定タスクの学習の枠組みのなかで不適当な事例を取り除く方法を示す (Algorithm 1).

はじめに並列構造の範囲同定モデル M [14, 17] を少量のラベル付きデータ L で学習し, 以降は L に加えて大量のラベルなしデータ U から生成された事例を用いて学習を行う. 各学習イテレーションにおいて, U から並列構造を持たない文 x をランダムに選択し, 提案手法の生成モデル G によって並列構造を含む文 x' とそれに対応する並列構造範囲のアノテーション y' を得る⁵⁾. 生成によって得た事例 (x', y') は学習モデル M によってスコアが割り当てられ, その値が閾値 δ 以上であれば適当な例だと見なしてミニバッチ B に追加する. この生成・選定の手順を事例が K 個得られるまで最大で K_{max} 回繰り返し, $N_{batch} - K$ 個の事例を L から B に加えて, M の学習・更新を行う.

4) 要素ごとに平均をとる (element-wise mean) 関数の場合, あるトークンに対する値が片方の logit で著しく高く, もう一方の logit ではそれほど高い値でない場合, 平均化しても依然として高い値となってしまう, 過剰適合が生じやすい.

5) 文の異なるスパンを参照することで異なる並列構造を生成できることから, 復元抽出によって文 x をサンプルする.

Algorithm 1 並列構造の生成と選定を統合した学習

Input: ラベル付き文集合 L ; ラベルなし文集合 U ; 範囲同定モデル M ; 生成モデル G

Constants: バッチサイズ N_{batch} ; 生成による事例数 $K (\leq N_{batch})$; 生成回数の上限 K_{max} ; 事例選定の閾値 δ

Output: 学習済み範囲同定モデル M

```
1: train  $M$  on  $L$ 
2: repeat
3:    $B \leftarrow \{\}$ 
4:    $k \leftarrow 0$ 
5:   while  $|B| < K$  and  $k < K_{max}$  do
6:      $x \leftarrow \text{SAMPLE}(U, 1)$ 
7:      $(x', y') \leftarrow G(x)$ 
8:     if  $\text{CALCScore}((x', y'), M) \geq \delta$  then
9:        $B \leftarrow B \cup \{(x', y')\}$ 
10:    end if
11:     $k \leftarrow k + 1$ 
12:  end while
13:   $B \leftarrow B \cup \text{SAMPLE}(L, N_{batch} - K)$ 
14:  update  $M$  on  $B$ 
15: until convergence
16: return  $M$ 
```

3 実験

本研究は提案手法による並列構造の生成を並列構造の範囲同定の学習に適用することで性能評価する⁶⁾.

3.1 実験設定

データ構築 並列構造がアノテーションされた Penn Treebank (PTB) [18] を用いて実験を行う。学習データの追加による性能向上を確かめるために、低リソース状況下における評価を行う。学習セット 39,832 文のうち並列構造を持つ文 15,481 の中から、ランダムに 300 文 (または 600 文) 選び、5:1 の比で学習用データ L とモデル選択のための検証用データに分割し、並列構造を持たない 24,351 文を生成元のデータ U として用いる⁷⁾。実験の評価は PTB のテストセットで行い、PTB の開発セットは追加分析のために用いる。同様の手順によって、生物医学系論文のアブストラクトから成る GENIA Treebank [19]

6) 定性的分析を Appendix C に示す。

7) 言語モデルに十分な文脈を与えるために、10 語未満から成る文は U から取り除く。

でも実験を行う。

評価方法とモデル 各等位接続詞に対する並列構造全体の始点・終点の同定精度 (accuracy) によってモデルを評価する。タスクに使用するモデルとして、Teranishi ら [14] の手法と同様に、並列構造の始点・終点の可能な全ての組み合わせについてスコア計算を行う機構を用いるが、下層には BERT (BERT-base-cased) を用いる。位置 k に出現する等位接続詞に対する並列構造の始点・終点ペア (i, j) のスコアは次のように計算される。

$$\begin{aligned} \text{SCORE}_\theta(i, j, k) &= \text{MLP}(F_\theta(i, j, k)) \\ F_\theta(i, j, k) &= [\mathbf{h}_i - \mathbf{h}_{k+1}; \mathbf{h}_j - \mathbf{h}_{k-1}] \end{aligned} \quad (2)$$

ここで MLP は 2 層の線形変換と ReLU の活性化関数から成るニューラルネットワークであり、 \mathbf{h}_t は BERT の Transformer から出力された t 番目のトークンの表現ベクトルである。可能な組み合わせの全スコアは softmax 関数によって正規化され、各スコアは 0 から 1 の値となる。このニューラルネットワークを学習データ L のみで訓練したものをベースラインのモデルとして定める。また、提案手法における文の参照スパンは、句構造解析によって得られた構文木の任意の句をランダムに選択することによって得る。学習の詳細設定は Appendix B に示す。

3.2 結果と分析

実験結果を表 1 に示す。提案手法による並列構造の生成によって、範囲同定モデルは概してベースラインよりも高い性能を達成した。このことは提案手法が BERT と T5 のモデルを用いてタスクの性能向上に寄与するよう有用な事例を生成していることを示唆している。また、T5 は BERT と比較するとより効果的な事例が生成できており、これは事前学習において BERT が独立したマスクに対して一つのトークンを予測しているのに対し、T5 は sequence-to-sequence 学習によってマスクに対してトークンの系列を予測しているという違いによるものと推測される。並列構造の統語範疇ごとの評価では、T5 を用いた生成によって S や SBAR など節の並列構造の同定精度が改善しており、T5 が比較的短いトークン系列を予測するよう事前学習されていることを考えると予想に反する結果である。一方で、名詞の並列構造はそれほど改善が見られないどころか精度が下がっているケースも観測される。これは少量の学習データ L に含まれる並列構造の多く

Penn Treebank							
#	Model	NP/NX	ADJP/ADVP	VP	PP	S/SBAR	Overall
250	baseline (no generation)	67.61	70.90	59.09	30.76	48.05	61.92 ± 0.93
250	+ generation by BERT	66.05	69.45	64.82	75.38	47.79	62.82 ± 1.27
250	+ generation by T5	67.73	69.09	66.73	76.92	55.59	65.44 ± 1.82
500	baseline (no generation)	74.45	81.45	70.85	44.61	63.14	71.19 ± 0.58
500	+ generation by BERT	71.65	76.72	69.34	86.15	60.12	69.24 ± 1.64
500	+ generation by T5	73.69	81.45	73.26	86.15	63.89	72.04 ± 0.54
GENIA Treebank							
250	baseline (no generation)	61.54	48.00	65.41	53.80	40.25	59.42 ± 1.75
250	+ generation by BERT	62.41	57.50	67.35	69.29	39.24	61.34 ± 2.73
250	+ generation by T5	65.72	56.00	68.38	70.14	40.50	63.92 ± 1.15
500	baseline (no generation)	67.12	63.50	70.19	74.64	44.05	66.05 ± 1.66
500	+ generation by BERT	68.86	64.00	71.48	74.64	53.16	68.08 ± 0.92
500	+ generation by T5	69.29	67.00	73.16	76.90	53.41	68.86 ± 2.13

表 1 生成された事例を用いた fine-tuning によるテストセットにおける結果. 最初の列は学習に使用したラベル付きデータのサイズを表す. 表中の数字はデータ構築・生成・fine-tuning を 5 回通じて試行した結果の平均と標準偏差を表す.

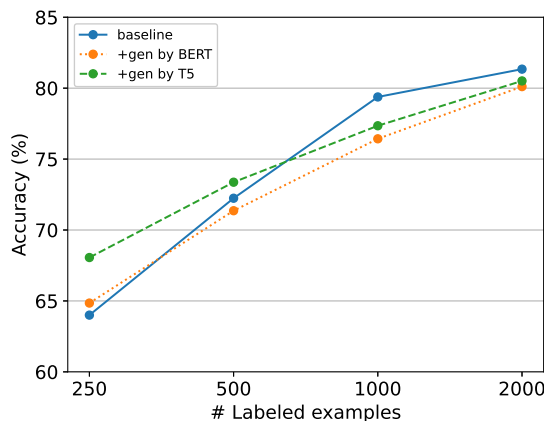


図 2 学習データのサイズによる精度の推移.

が名詞の並列であり, 名詞並列に関する学習が十分にできているため, 生成による学習データのかさ増しがかえって悪影響を及ぼしていると考えられる. 学習データ L のサイズによるパフォーマンスの推移を PTB の開発セットにて観測すると, 学習データが増えるにつれて提案手法による改善幅は小さくなり, 十分な学習データが利用できる状況になるとむしろ精度の低下につながっている (図 2).

追加の分析として, 生成結果の選定や同期的生成を適用しない場合の性能を PTB の開発セットにて調査した. 前者は選定の閾値 δ を 0 に設定したものであり, 後者は文 $S^{(1)}$ のみを生成に用いたものである. ただし, 同期的生成を用いない後者の設定は範囲が不適当な並列構造が生成されやすく, モデル M による選定の過程で取り除かれてしまうため, 提案手法によって得られた文 x' のみを採用して並列構造の範囲 y' はモデル M が自身で予測し, その確信

Setting	Overall
baseline	64.00
+ generation by BERT	64.85
– filtering	63.06
– sync + self-train	63.06
+ generation by T5	68.06
– filtering	63.25
– sync + self-train	66.83

表 2 異なる設定による精度の違い ($|L| = 250$).

度 (確率値) に基づいて選定を行う (self-training). 実験結果を表 2 に示す. 生成された並列構造を選定せずに使用した場合に性能を大きく損なうことから, 提案手法が不適当な並列構造を一定の割合で生成しており, 一方で選定の仕組みによって学習のノイズとなるような事例を効果的に取り除いていることがわかる. また同期的生成を適用せずに self-training によって範囲を割り当てた場合も性能の低下につながった. モデル M が誤って予測した範囲によって学習が進むにつれてノイズが増幅しているのに対し, 提案手法による同期的生成では学習進度に関わらず適当な並列構造を一定の割合で生成できていることが示唆される.

以上の分析を踏まえて, 生成される並列構造とその範囲が適当である事例をより高い割合で生成できるよう手法を改善していくことが提案手法の今後の課題である⁸⁾.

8) そのような方法として, T5 や BERT に並列構造の生成を促すような fine-tuning を適用することが考えられる.

謝辞

本研究の一部は、JST・AIP 日独仏 AI 研究 (JP-MJCR20G9) の支援を受けたものである。

参考文献

- [1] 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)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [2] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.
- [3] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, **Advances in Neural Information Processing Systems**, Vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.
- [5] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 4222–4235, Online, November 2020. Association for Computational Linguistics.
- [6] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 3816–3830, Online, August 2021. Association for Computational Linguistics.
- [7] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional bert contextual augmentation. In João M. F. Rodrigues, Pedro J. S. Cardoso, Jânio Monteiro, Roberto Lam, Valeria V. Krzhizhanovskaya, Michael H. Lees, Jack J. Dongarra, and Peter M.A. Sloot, editors, **Computational Science – ICCS 2019**, pp. 84–95. Springer International Publishing, 2019.
- [8] Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. Soft contextual data augmentation for neural machine translation. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 5539–5544, Florence, Italy, July 2019. Association for Computational Linguistics.
- [9] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do not have enough data? deep learning to the rescue! In **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, pp. 7383–7390, Apr. 2020.
- [10] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. In **Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems**, pp. 18–26, Suzhou, China, December 2020. Association for Computational Linguistics.
- [11] Gözde Gül Şahin and Mark Steedman. Data augmentation via dependency tree morphing for low-resource languages. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 5004–5009, Brussels, Belgium, October–November 2018. Association for Computational Linguistics.
- [12] Haoyue Shi, Karen Livescu, and Kevin Gimpel. Substructure substitution: Structured data augmentation for NLP. In **Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021**, pp. 3494–3508, Online, August 2021. Association for Computational Linguistics.
- [13] Jessica Ficlcr and Yoav Goldberg. A neural network for coordination boundary prediction. In **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 23–32, Austin, Texas, November 2016. Association for Computational Linguistics.
- [14] Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. Coordination boundary identification with similarity and replaceability. In **Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 264–272, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.
- [15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, Vol. 21, No. 140, pp. 1–67, 2020.
- [16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.
- [17] Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. Decomposed local models for coordinate structure parsing. 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)**, pp. 3394–3403, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [18] Jessica Ficlcr and Yoav Goldberg. Coordination annotation extension in the Penn Tree Bank. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 834–842, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [19] Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun’ichi Tsujii. Syntax annotation for the GENIA corpus. In **Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts**, 2005.
- [20] Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 3499–3505, Florence, Italy, July 2019. Association for Computational Linguistics.

1. Accepted	We're sorry to report that on Monday President Bush accepted the resignation of William Allen as chairman <u>[of the U.S. Civil Rights Commission]</u> and <u>[of the U.S. Department of Justice]</u> .
2. Accepted	The opposition charged that the money was used <u>[to bribe Indian government officials]</u> and <u>[to support the Congress]</u> , an allegation denied by Mr. Gandhi's administration.
3. Rejected	" <u>[I wouldn't say it's quite a veto]</u> and <u>[it's not a veto]</u> ," Mr. Boren demurs.
4. Rejected	Most analysts had expected a sharper decline <u>[after the steep rise in August]</u> and <u>[earlier in this year]</u> .

表3 提案手法によって生成された並列構造を含む文とその範囲の例。下線部は生成に使われた参照スパンを表す。

A BERT を用いた同期的生成

BERT を用いた同期的テキスト生成では、文 $S^{(1)}$, $S^{(2)}$ の空所は BERT で用いられている [MASK] トークンの系列によって作られる。T5 とは異なり、生成されるトークン数は [MASK] の数に依存するが、ここでは参照スパンと同じトークン数を使用する。文 $S^{(1)}$, $S^{(2)}$ は [SEP] トークンによって連結されて BERT に与えられ、二つの [MASK] トークン系列に対する予測は次式に従って行われる。

$$\begin{aligned} \mathbf{u}'_{l+t} &= \mathbf{u}'_{m+t} = f(\mathbf{u}_{l+t}, \mathbf{u}_{m+t}) \\ T'_{t+1} &= \arg \max(\mathbf{u}'_{l+t}) \\ T'_{t+1} &= \arg \max(\mathbf{u}'_{m+t}) \end{aligned} \quad (3)$$

ここで $t (\geq 0)$ はマスク箇所の始点 l , m からの相対位置である。

B 学習方法の設定

各ニューラルネットワークモデルに対して 10,000 イテレーションの学習を行い、構築した検証用データにて最も良い精度 (accuracy) を達成したモデルをテストセットでの評価に用いる。ベースラインがラベル付きデータ L のみで学習されるのに対し、提案手法のモデルは最初の 1,000 イテレーションを L で学習し (Algorithm 1 の 1 行目)、残りの 9,000 イテレーションはラベルなしデータ U から生成された事例と L によって学習を行う (Algorithm 1 の 2-15 行目)。全ての実験において、提案手法のハイパーパラメータとして $N_{batch} = 16$, $K = 8$, $K_{max} = 16$, $\delta = 0.7$ を設定する。並列構造の生成には BERT-large-cased と T5-large のモデルを用い

る。参照スパンは Berkeley neural parser [20] によって得られた構文木に含まれる句をランダムに選択することで得るが、選択対象の句は NP, VP, ADJP, ADVP, PP, SBAR, S の句 (文全体を表す root ノードは除く) に限定される。学習において AdamW を学習率 $2e-5$ で適用し、学習率はウォームアップなしで学習進度に合わせて線形に減少させる。また、勾配をノルムの閾値 1.0 でクリッピングし、weight decay を 0.01 の強度で適用する。モデルは 100 イテレーションごとに評価され、1,000 イテレーション以降に 1,000 イテレーション連続で性能向上が見られない場合は学習を早期に打ち切る。式 2 の MLP は 256 ユニットから成る一層の隠れ層を持ち、dropout を確率 0.5 で適用する。並列構造の範囲としては単語の境界のみを候補として、単語が複数のトークンから成る場合は最後のトークンの表現を用いてスコア計算を行う。

C 生成された並列構造の事例分析

T5 を用いた提案手法によって生成された並列構造の事例を表 3 に示す。例 1 と例 2 はそれぞれ、生成モデルが前置詞句・動詞句を参照して生成した並列構造を表しており、これらの並列構造とその範囲は範囲同定モデル M によって適当なものとして正しく判定されている。対して例 3 は、参照スパンが複文であるために生成された並列構造に対してその範囲が不適当であるが、この事例はモデル M によって学習から取り除かれている。例 4 では生成モデルが副詞句の並列構造を生成しているが、モデル M によって高い確度が割り当てられず不適当な事例として判定されている。