細分化された局所的モデルによる並列構造の構文解析

寺西 裕紀　　進藤 裕之　　松本 裕治
奈良先端科学技術大学院大学 先端科学技術研究科
{teranishi.hiroki.sw5, shindo, matsu}@is.naist.jp

1 はじめに

並列構造とは等位接続詞などが結びつける句（並列句）から成る構造である．並列構造の範囲の解釈には曖昧性があり，並列構造は文の意味理解を困難にさせるばかりでなく，構文解析の誤りの原因となっている．これまで並列構造解析は，並列句の類似性に着目した手法 [5, 4, 7, 3]，類似性と可換性に着目した手法 [2, 9, 8] が研究されてきた．Ficler らの手法 [2] や寺西らの手法 [8] ではニューラルネットワークを用いて高い解析性能を達成しているが，三つ以上の並列句から成る並列構造の解析や文中の複数の並列構造の同時解析については限定的にしか取り扱えない．

本研究では，句のペアが並列となる場合に高いスコアを出力するスコア関数を学習し，解析時にはスコア関数を用いて構文解析を行うというフレームワークを提案する．提案手法では並列構造の範囲同定のタスクを，等位接続詞の同定，句のペアの内側の境界の同定，句のペアの外側の境界の同定の三つのサブタスクに分け，それぞれのサブタスクを学習した三つのニューラルネットワークによってスコア関数を構成する．解析時には範囲の競合なく並列構造を同定するために，並列構造の導出規則を用いた CKY 構文解析の手法 [3] を適用する．英語の並列構造を対象にした評価実験において，既存研究を上回る精度が得られたことを示す．

2 提案手法

本研究で提案するフレームワークの概要を図 1 に示す．N 語の系列 $w_{1:N} = w_1, \ldots, w_N$ と対応する品詞系列 $p_{1:N} = p_1, \ldots, p_N$ から成る文 S を入力として，タスクの出力として並列構造の集合 $\{\langle c, \{[b_1, e_1], \ldots, [b_n, e_n]\}\rangle | S\} (n \geq 2)$ が期待される．ここで c は等位接続詞，$[b_k, e_k]$ は b_k から e_k 番目の語をスパンとする並列句である．並列構造の数や各々の並列構造に含まれる並列句の数は文の表層からは判定できないが，等位接続詞やカンマを手がかりとして並列句のペアを見つけることができる．そこで提案手法では文に含まれる並列構造の同定を，並列句のペアの同定と置き換えてタスクの学習を行う．

$$\begin{aligned} X &= \{w_{1:N}, p_{1:N}, C\} \\ C &= \{t | w_t \in \mathbb{S}_{cc} \cup \mathbb{S}_{sub\text{-}cc}\} \\ Y &= \{\langle y_t^{ckey}, y_t^{pair}\rangle | t \in C\} \end{aligned} \quad (1)$$

説明のため，等位接続詞に伴って並列句を連接させる働きのある語を準等位接続詞と呼び，等位接続詞または準等位接続詞となり得る語を並列キーと呼ぶ．式 1 において，y_t^{ckey} は並列キー w_t が（準）等位接続詞であれば $y_t^{ckey} = 1$，そうでなければ $y_t^{ckey} = 0$ となる．また，y_t^{pair} は並列キー w_t が結びつける並列句スパンのペアであり，$y_t^{ckey} = 0$ のとき $y_t^{pair} = \varnothing$ である．本研究では，解析対象の等位接続詞 $\mathbb{S}_{cc}$ と準等位接続詞 $\mathbb{S}_{sub\text{-}cc}$ をそれぞれ {“and”, “or”, “but”, “nor”, “and/or”} と {“,”, “;”, “:”} とする．

本研究で提案するニューラルネットワークは等位接続詞分類器，内側境界スコア付与モデル，外側境界スコア付与モデルの三つから成り，式 1 に示したタスクをそれぞれ部分的に解くことで学習する．解析時には学習した三つのモデルを用いて確率変数 y_t^{ckey}，y_t^{pair} に対する確率分布を求め，文脈自由文法規則を用いた CKY 構文解析により並列構造を表す構文木を導出し，構文木から最終的に並列構造の集合を取り出す．

2.1 モデル

エンコーダ

三つのモデルは下層に共通のエンコーダを持ち，エンコーダから出力されたベクトル系列がそれぞれのモデルで用いられる．エンコーダは単語と品詞の系列を文レベルのベクトル表現の系列に変換する．

$$\mathbf{h}_{1:N} = \mathrm{BiLSTMs}(f_{input}(w_{1:N}, p_{1:N})) \quad (2)$$

ここで BiLSTMs は d^{hidden} 次元ベクトルを隠れ状態に持つ Long Short-Term Memory による演算を系列の双方向から適用するニューラルネットワークである．本研究では f_{input} として，単語の分散表現，品詞の分散表現，単語を構成する文字列のベクトル表現の三つのベクトルを連結したものを出力するニューラルネットワークを用いる．文字列のベクトル表現の計算には畳み込みニューラルネットワーク（CharCNN）[6] を用いる．

等位接続詞分類器

等位接続詞分類器は並列キーが（準）等位接続詞であるかどうかを二値分類する．

$$f_{ckey}(w_t) = \mathbf{W}^{ckey}\mathbf{h}_t + \mathbf{b}^{ckey} \quad (3)$$

$$P(y_t^{ckey} | w_t, \theta) = \mathrm{softmax}(f_{ckey}(w_t)) \quad (4)$$

$$\ell_\theta^{ckey}(X, Y) = -\sum_{\langle y_t^{ckey}, y_t^{pair}\rangle \in Y} \log P(y_t^{ckey} | w_t, \theta) \quad (5)$$

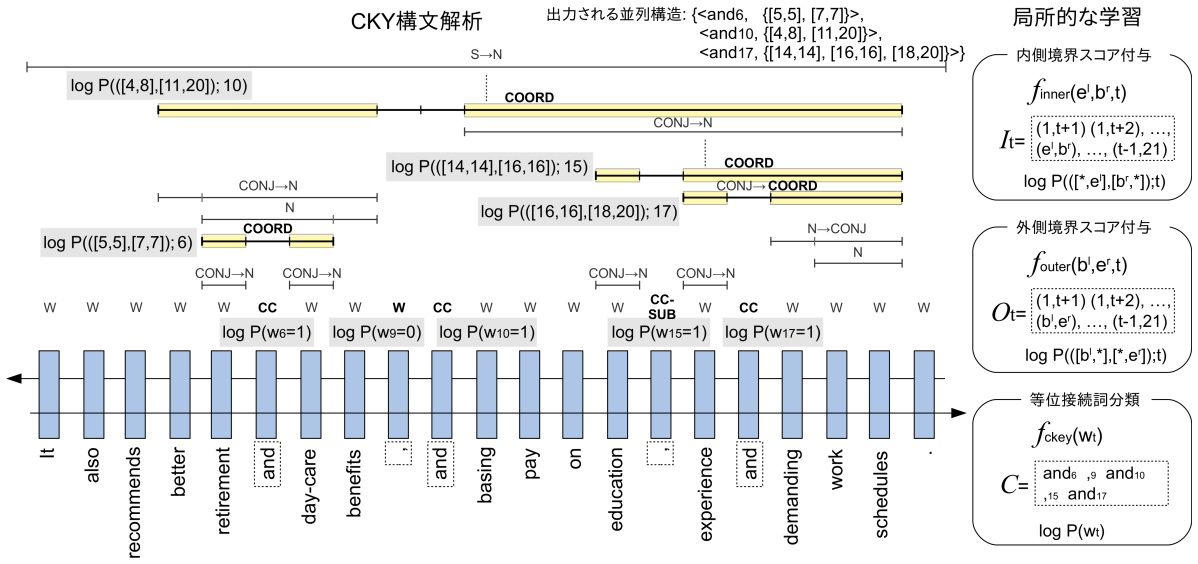

図 1: 提案するフレームワークの概要.

ここで，$\mathbf{W}^{ckey} \in \mathbb{R}^{2\times 2d^{hidden}}$ と $\mathbf{b}^{ckey} \in \mathbb{R}^2$ はモデルパラメータである．

内側境界スコア付与モデル

並列キーが等位接続の働きをするとき，並列キーによって結びつけられる左右の句のペアの始点・終点をそれぞれ b^l, e^l, b^r, e^r と表す．本研究では左側の句の終点と右側の句の始点を内側境界，左側の句の始点と右側の句の終点を外側境界と呼ぶ．内側境界スコア付与モデルは句のペアの内側境界に基づいてスコアを計算するモデルである．並列キー w_t における句のペアの内側境界 (e^l, b^r) は次のように計算される．

$$f_{inner}(e^l, b^r, w_t) = \mathbf{w}_2^{in}\ \mathrm{ReLU}(\mathbf{W}_1^{in}[\mathbf{h}_{e^l}; \mathbf{h}_{b^r}] + \mathbf{b}_1^{in}) + \mathrm{b}_2^{in} \tag{6}$$

$$\textsc{Score}_\theta^{inner}(e^l, b^r, w_t) = f_{inner}(e^l, b^r, w_t) \tag{7}$$

ここで，$\mathbf{W}_1^{in} \in \mathbb{R}^{d^{in}\times 4d^{hidden}}$, $\mathbf{b}_1^{in} \in \mathbb{R}^{d^{in}}$, $\mathbf{w}_2^{in} \in \mathbb{R}^{d^{in}}$, $\mathrm{b}_2^{in} \in \mathbb{R}^1$ はモデルパラメータである．内側境界の全ての可能な組み合わせについてスコアを計算することで内側境界の確率を得る．

$$I_{w_t} = \{(1, t+1), (1, t+2), \ldots, (1, N), (2, t+1), \ldots, (t-1, N)\} \tag{8}$$

$$P(y_t^{pair} = ([*, e^l], [b^r, *])|w_t, \theta) = \frac{\exp\left(\textsc{Score}_\theta^{inner}(e^l, b^r, w_t)\right)}{\sum_{(e'^l, b'^r)\in I_{w_t}} \exp\left(\textsc{Score}_\theta^{inner}(e'^l, b'^r, w_t)\right)} \tag{9}$$

$$\ell_\theta^{inner}(X, Y) = -\sum_{\langle y_t^{ckey}, y_t^{pair}\rangle \in Y} y_t^{ckey} \log P(y_t^{pair}|w_t, \theta) \tag{10}$$

項 $y_t^{ckey} \log P(y_t^{pair}|w_t, \theta)$ は $y_t^{pair} = 1$ のときにクロスエントロピー損失となり，$y_t^{pair} = 0$ の場合は 0 となる．

外側境界スコア付与モデル

外側境界スコア付与モデルは次式で定義される．

$$f_{outer}(b^l, e^r, w_t) = \mathbf{w}_2^{out}\ \mathrm{ReLU}(\mathbf{W}_1^{out}\ \mathbf{r}) + \mathbf{b}_1^{out}) + \mathrm{b}_2^{out} \tag{11}$$

$$\mathbf{r} = \left[\mathbf{h}_{b^l} - \mathbf{h}_{t+1}; \mathbf{h}_{e^r} - \mathbf{h}_{t-1}\right]$$

$$\textsc{Score}_\theta^{outer}(b^l, e^r, w_t) = f_{outer}(b^l, e^r, w_t) \tag{12}$$

ここで，$\mathbf{W}_1^{out} \in \mathbb{R}^{d^{out}\times 4d^{hidden}}$, $\mathbf{b}_1^{out} \in \mathbb{R}^{d^{out}}$, $\mathbf{w}_2^{out} \in \mathbb{R}^{d^{out}}$, $\mathrm{b}_2^{out} \in \mathbb{R}^1$ はモデルパラメータである．内側境界スコア付与モデルと同様に，外側境界の集合 O_{w_t}，外側境界の確率 $P(y_t^{pair} = ([b^l, *], [*, e^r])|w_t, \theta)$，損失関数 ℓ_θ^{outer} を定義する．なお，I_{w_t} と O_{w_t} は等しい集合となる．

学習

エンコーダ及び三つのモデルのパラメータ集合 θ は次の損失関数を最小化することにより学習する．

$$L(\theta) = \sum_{(X,\hat{Y})\in D} \left(\ell_\theta^{ckey}(X, \hat{Y}) + \ell_\theta^{inner}(X, \hat{Y}) + \ell_\theta^{outer}(X, \hat{Y})\right) \tag{13}$$

D は文とその並列構造の対の集合から成る学習データである．

2.2 CKY 構文解析

三つのモデルによる並列句ペアの予測は並列キーごとに独立して行われるが，複数の並列構造は範囲が一部分だけ重なり合うことはなく，入れ子になるか全く重なり合わないかという制約がある．また，準等位接続詞によって結びつけられた並列句ペアは等位接続詞によって形成された並列構造に含まれなければならない．これらの制約下で並列

構造の範囲の組み合わせを探索するために，表 1 に示す並列構造の導出規則を用いる[1]．表 1 の規則を用いた CKY 構文解析によって，最もスコアの高い並列構造の組み合わせが構文木として出力される．構文木と並列構造は一対一の関係にあり，相互に変換できる[2]．

CKY 構文解析の過程において COORD と前終端記号 CC，CC-SUB，W のノードのみにスコアを付与する．語 $w_k \in \mathbb{S}_{cc} \cup \mathbb{S}_{sub\text{-}cc}$ から導出される CC と CC-SUB には $\log P(w_k = 1)$ を，W には $\log(P(w_k = 0))$ を割り当て，並列キーとならない語から導出される W に対しては 0 を付与する．COORD に対しては，CC によって結びつけられる左右の CONJ ペアのスパンを用いて，スコア $\log P(([i,j],[l,m])) = \log P(([*,j],[l,*])) + \log P(([i,*],[*,m]))$ を付与する．規則 (2) においても同様に CC-SUB の左右に隣接する CONJ，すなわち左隣の CONJ と子 COORD の最左 CONJ をペアとしてスコアを割り当てる．単語数 N の文において最もスコアの高い構文木は動的計画法により時間計算量 $\mathcal{O}(N^3)$ で求められる．

3 実験

3.1 実験設定

並列構造のアノテーションが付与された Penn Treebank[1] を用いる．また，先行研究 [2, 8] とは異なり，データセットから二重引用符 “と” を全て取り除く．これは英語において，引用符を伴う並列句が $\langle \ldots$ *“Daybreak,” “Daywatch,” “Newsday,” and “Newsnight,”* $\ldots \rangle$ のように変形されるためである．単語の分散表現には GloVe[3] を用い，品詞タグは Stanford POS Tagger[4] の 10 分割ジャックナイフ法によって付与し，品詞の分散表現は標準正規分布からランダムに初期化した．モデルパラメータの最適化は Adam を用いて確率的勾配降下法によって行った．実験で使用した最終的なハイパーパラメータは次のとおりである．単語ベクトル（GloVe）の次元 : 100，品詞ベクトルの次元 : 50，文字ベクトルの次元 : 10，CharCNN のウィンドウ幅 : 5，文字列ベクトルの次元 : 50，LSTM の隠れ状態ベクトルの次元 d^{hidden} : 512， BiLSTMs の層 : 2，内側境界スコア付与モデルの中間層ユニット d^{in} : 1024，外側境界スコア付与モデルの中間層ユニット d^{out} : 1024，Dropout（単語ベクトル，品詞ベクトル，文字列ベクトル，LSTM の隠れ状態ベクトル，内側境界スコア付与モデルの中間層，外側境界スコア付与モデルの中間層に適用）の比率 : 0.5，学習率の初期値 : 0.001，勾配クリッピングの閾値 : 5.0.

比較実験のベースラインとして寺西らの提案モデル [8] を再実装し，さらに解析時の拡張を施したものを用いた（以降，*Teranishi+17ext* と呼ぶ）．寺西らの手法では並列構造全体の始点・終点を学習・予測し，解析時には並列構造をカンマで区切ることによって個々の並列句を取り出している．また，並列構造の範囲は個々に予測されるため，複数の並列構造の範囲が競合する場合がある．そこで解析時には文内の全ての並列構造の範囲が競合せずに最もスコアが高くなる組み合わせを出力するよう拡張する[5]．ベースラインモデルのエンコーダ部分については本提案手法と同様のモデルを使用する．

手法の評価は並列構造の範囲の一致における適合率・再現率・F 値によって行う．並列構造の範囲の一致基準は寺西ら [8] と同様に次の四つを用いる．

- **whole**: 並列構造の始点と終点の一致．
- **outer**: 並列構造の最初と最後の並列句の範囲の一致．
- **inner**: 並列構造の等位接続詞前後の並列句範囲の一致．
- **exact**: 並列構造の全並列句範囲の一致．

また，提案手法の有効性を検証するため，文単位での並列構造の完全一致率についても評価を行う．

非終端記号

COORD	Coordination; 並列構造
CONJ	Conjunct; 並列句
CC	Coordinating conjunction; 等位接続詞
CC-SUB	Sub-coordinator; 準等位接続詞
W	Word; 語
N	Non-coordination; 非並列構造
S	Sentence; 文

並列構造に関する規則

(1)	COORD	→	CONJ N? CC N? CONJ
(2)	COORD	→	CONJ CC-SUB COORD
(3)	CONJ	→	COORD
(4)	CONJ	→	N

並列構造以外の規則

(5)	S	→	COORD
(6)	S	→	N
(7)	N	→	COORD N
(8)	N	→	W COORD
(9)	N	→	W N
(10)	N	→	W

前終端記号の規則

(11)	CC	→	(and|or|but|nor|and/or)
(12)	CC-SUB	→	(,|;|:)
(13)	W	→	*

表 1: 並列構造を表す構文木の導出規則．(. . . |. . .) は括弧内のいずれかの語に，“*” は任意の語にマッチし，“?” は直前の要素の 0 または 1 回の出現を表す．

3.2 実験結果

並列構造単位での評価を表 2 に示す．提案手法は全ての一致基準においてベースラインの性能を上回った．ベースラインのモデルは並列構造の範囲を決定した後に個々の並列句に分割していることから inner より whole の精度が高

[1] 原ら [3] の規則と異なり，等位接続詞と並列句の間に副詞句などの句がある場合でも並列構造を導出できる．

[2] Penn Treebank[1] にアノテーションされている並列構造のうち，99.5%が表 1 の規則により導出可能である．

[3] https://nlp.stanford.edu/projects/glove/

[4] https://nlp.stanford.edu/software/tagger.shtml

[5] 並列構造内の並列句に関する範囲の制約は設けない．

		Development						Test					
		All			NP			All			NP		
		P	R	F	P	R	F	P	R	F	P	R	F
Ours	whole	78.60	78.41	78.51	79.26	78.71	78.98	76.88	77.16	77.02	78.75	78.50	78.62
	outer	77.18	77.00	77.09	78.57	78.03	78.30	75.33	75.61	75.47	77.95	77.70	77.83
	inner	79.19	79.00	79.10	80.64	80.09	80.36	77.60	77.88	77.74	80.19	79.93	80.06
	exact	**76.95**	**76.76**	**76.85**	**78.11**	**77.57**	**77.84**	**75.33**	**75.61**	**75.47**	**77.95**	**77.70**	**77.83**
Teranishi+17ext	whole	78.78	77.94	78.36	78.52	77.80	78.16	77.36	76.52	76.94	78.72	78.34	78.53
	outer	74.49	73.70	74.09	76.67	75.97	76.32	72.03	71.24	71.63	75.36	75.00	75.17
	inner	76.04	75.23	75.63	77.82	77.11	77.47	74.14	73.33	73.74	77.44	77.07	77.25
	exact	74.13	73.34	73.74	76.21	75.51	75.86	71.48	70.70	71.08	75.20	74.84	75.01
Ficler+16*	inner	72.34	72.25	72.29	75.17	74.82	74.99	72.81	72.61	72.7	76.91	75.31	76.1

表 2: 並列構造単位での各一致基準による評価．“Ficler+16”[2] では引用符に関する前処理について言及されておらず，厳密な比較はできない．

Model	Sentence	Development	Test
Ours	All	489 / 673 = **72.65**	619 / 873 = **70.90**
	- Simple	378 / 481 = **78.58**	476 / 609 = **78.16**
	- Complex	111 / 192 = **57.81**	143 / 264 = **54.16**
	- Consecutive	41 / 66 = **62.12**	56 / 96 = **58.33**
	- Multiple	79 / 146 = **54.10**	96 / 197 = **48.73**
Teranishi +17ext	All	468 / 673 = 69.53	577 / 873 = 66.09
	- Simple	358 / 481 = 74.42	444 / 609 = 72.90
	- Complex	110 / 192 = 57.29	133 / 264 = 50.37
	- Consecutive	40 / 66 = 60.60	48 / 96 = 50.00
	- Multiple	78 / 146 = 53.42	92 / 197 = 46.70

表 3: 文単位の並列構造の完全一致率．

いのに対し，提案手法は並列キー前後の並列句からボトムアップに並列構造を構築していることから whole より inner の精度が高い．文単位での並列構造の一致についての評価結果を表 3 に示す．並列構造を含む文を次のとおりに分類して分析を行っている．

- **All**：並列構造を持つ全ての文．
- **Simple**：二つの並列句から成るただ一つの並列構造を持つ文．
- **Complex**: Consecutive または Multiple に分類される文．
- **Consecutive**: 三つ以上の並列句から成る並列構造を少なくとも一つ持つ文．
- **Multiple**: 複数の並列構造を持つ文．

提案手法はいずれの文においてもベースラインの精度を上回った．特に Simple の文での精度向上が全体の精度向上に寄与している．また，既存手法の課題となっていた Consecutive と Multiple の文においても，提案手法はベースラインより高い解析性能を発揮した．ベースラインでは個々の並列句の範囲は学習されておらず，並列構造の範囲を分割することで並列句範囲を決定しているため，真に準等位接続詞にならないようなカンマによって並列構造を誤った並列句に分割するというエラーが起こる．対して，提案手法は等位接続詞前後の並列句だけでなく準等位接続詞前後の並列句についても学習を行っており，三つ以上の並列句を持つ並列構造をより正確に導出できると考えられる．

4 おわりに

本稿では，並列構造解析のためのフレームワークを提案した．三つのサブタスクを解くニューラルネットワークを用いてスコア関数を学習し，解析時には並列構造の導出規則を用いた CKY 構文解析により並列構造の範囲制約を満たす最適な組み合わせを出力する．実験の結果，提案手法は既存手法 [8] を拡張したベースラインを上回る解析性能を達成し，複雑な並列構造の導出が可能となった．今後は k-best 構文解析の結果を言語モデルを用いてリランキングするなど，より高次の情報を取り入れて解析性能の向上を目指すとともに，異なるドメインや日本語を対象として手法を応用する．

謝辞 本研究は JST CREST（課題番号: JPMJCR1513）の支援を受けて行った．

参考文献

[1] Jessica Ficler and Yoav Goldberg. Coordination annotation extension in the penn tree bank. In *ACL*, pp. 834–842, 2016.

[2] Jessica Ficler and Yoav Goldberg. A neural network for coordination boundary prediction. In *EMNLP*, pp. 23–32, 2016.

[3] Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. Coordinate structure analysis with global structural constraints and alignment-based local features. In *ACL-IJCNLP*, pp. 967–975, 2009.

[4] Deirdre Hogan. Coordinate noun phrase disambiguation in a generative parsing model. In *ACL*, pp. 680–687, 2007.

[5] Sadao Kurohashi and Makoto Nagao. A syntactic analysis method of long japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, pp. 507–534, 1994.

[6] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, pp. 1064–1074, 2016.

[7] Masashi Shimbo and Kazuo Hara. A discriminative learning model for coordinate conjunctions. In *EMNLP-CoNLL*, pp. 610–619, 2007.

[8] Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. Coordination boundary identification with similarity and replaceability. In *IJCNLP*, pp. 264–272, 2017.

[9] 山腰貴大, 大野誠寛, 小川泰弘, 中村誠, 外山勝彦. ニューラル言語モデルを用いた法令文の並列構造解析. 言語処理学会第 23 回年次大会, pp. 278–281, 2017.