Vol.2017-NL-232 No.5
2017/7/19

ニューラルネットワークに基づく並列句表現の学習と構造解析

寺西 裕紀[1,a)]　進藤 裕之[1,b)]　松本 裕治[1,c)]

概要：並列句解析の主たるタスクは複数の並列する句の範囲を同定することである．並列構造は文の構文・意味の解析において有用な特徴となるが，並列構造の曖昧性を解消する決定的な手法は現在においても確立されておらず，構文解析の誤りの主要な原因となっている．既存の手法では構文解析の結果や人手で設計された類似度の素性を用いており，パイプライン処理に起因する誤り伝搬や素性設計のコストが問題となっている．本研究では，近年自然言語解析に広く使用されているリカレントニューラルネットワークを用いて，構文解析の結果を用いずに単語の表層形と品詞情報のみから並列句の候補の表現を学習し，並列句の類似性と可換性の特徴に基づいて並列構造の範囲を予測する手法を提案する．Penn Treebank と GENIA コーパスを用いた実験の結果，先行研究を上回る解析精度を得た．

1. はじめに

並列構造は自然言語構造解析を困難にしている主たる要因である．近年，句構造や依存構造の解析手法は顕著に発展してきているが，並列構造を高い精度で解析する決定的な手法は確立されていない．並列構造の曖昧性が解消されることで構文解析の誤りを減らすだけではなく，科学技術論文の解析や文の要約，翻訳など広い範囲のアプリケーションでの利用が期待される．並列句の範囲を同定するタスクにおいて，従来の研究では並列構造の重要な手がかりのうち，並列句の候補となる句のペアの類似度に基づくモデルが提案されてきた [14], [4], [3]. しかしながら従来手法では類似度の計算に構文情報やシソーラスを用いて人手で設計された素性を利用しており，素性設計のコストや外部リソースの調達コストの点で問題がある．一方で並列句の可換性に着目した手法も提案されているものの，外部の構文解析器の出力を利用しており，解析器の誤りに派生する誤り伝搬や解析速度の点で課題が残っている [2]．また，従来手法では並列句の範囲の候補を抽出する際に文脈から切り離されて独立した句として比較されており，前後の文脈情報を考慮した句の特徴が用いられていない．

本研究では近年自然言語の解析で広く用いられている双方向型リカレントニューラルネットワークを使用して文脈情報を考慮した並列句のベクトル表現を抽出し，範囲の候補となる句の対からスコアを計算するモデルを提案する．提案モデルでは構文解析の結果やシソーラスなどの外部リソースを利用せず，単語の表層形と品詞情報のみから並列句の特徴を学習し，解析対象の等位接続詞に対して並列構造の範囲・非存在を判定する．

英語の並列句の範囲同定のタスクにおいて，Penn Treebank と GENIA コーパスを用いた先行研究との比較実験で既存手法を上回る結果を得たことを示す．

[1] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
[a)] teranishi.hiroki.sw5@is.naist.jp
[b)] shindo@is.naist.jp
[c)] matsu@is.naist.jp

2. 並列構造解析

本節では並列構造の特徴とタスクの詳細について述べる．

2.1 並列構造の特徴

並列構造解析は主に句を連接する働きのある等位接続詞などの語（本稿では「**並列キー**」と呼ぶ）によって結び付けられる並列句の範囲を当てることを指す．並列キーによって結び付けられる句の範囲は構文上一意に決まる場合があるが，文の意味や前後の文脈から決定づけられることが多い．しかしながら並列構造解析の困難さは並列範囲の曖昧性のみならず，次の２つの性質にも起因している（**図 1**）．

(i) １つの並列キーに対して連接する句は必ずしも並列キーの前後の２つの句に限定されず，３つ以上の句を伴う場合がある．
(ii) 文中に複数の並列構造が現れる場合があり，さらには一方の並列構造が他方の並列構造の句に含まれるような入れ子構造となるケースがある．

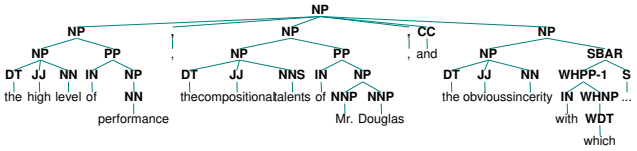
It was not an unpleasant evening, certainly, thanks to [the high level of performance], [the compositional talents of Mr. Douglas], and_{25} [the obvious sincerity with which Mr. Stoltzman chooses his selection].

(i) 3 つ以上の並列句が連接する例

Aside from [the Soviet economic plight] and_7 [talks on cutting (strategic) and_{12} (chemical) arms], one other issue the Soviets are likely to want to raise is naval force reduction.

(ii) 並列構造が入れ子となる例

図 1: 並列構造の範囲の性質の例



(a) 並列句の類似性の例

1. Aside from [the Soviet economic plight], one other ...
2. Aside from [talks on cutting (strategic) arms], one other ...
3. Aside from [talks on cutting (chemical) arms], one other ...

(b) 並列句の可換性の例

図 2: 並列句の特徴

これらの並列構造の範囲の性質に対して，並列する各々の句は次のような特徴を持つ．

(a) **類似性**：同一の並列構造に属する並列句は，句の構文構造・意味の点で類似性を持つ．

(b) **可換性**：同一の並列構造に属する並列句は，互いに入れ替えても文の流暢性が保たれる．

類似性について図 1 (i) の例では，and_{25}[*1]に対する並列句は全て名詞句 (NP) となっており，いずれも冠詞 (DT)，形容詞 (JJ)，名詞 (NN, NNS)，前置詞 (IN) という類似の構造が見られる (**図 2** (a))．また可換性については，図 1 (ii) の例の並列句を，”Aside from [talks on cutting (chemical) and_{12} (strategic) arms] and_7 [the Soviet economic plight], one other issue ...”と入れ替えても構文上誤りのない文として成立している．可換性の性質によって並列句の前後の句との流暢性を損なうことなく，各々の並列句について独立した文として展開することができる（図 2 (b))．

2 つの特徴は並列句の範囲を同定する強力な手がかりとなり得るが，下記の例に見られるように並列構造の全てで普遍的に利用できる特徴ではない（文と文の並列による非類似：”[The value of the two transactions wasn't disclosed], *but* [an IFI spokesman said no cash would change hands].”，語の省略による非可換：”[Honeywell's contract totaled $69.7 million], *and* [IBM's $68.8 million].”)．

本研究では並列句の類似性と可換性の両方に着目し，人手による素性設計ではなくニューラルネットワークを用いて，最適な特徴の重みを学習する．

2.2 タスクの定義

並列句の範囲の曖昧性解消のタスクの定義について述べる．文に現れる並列キーに対して，並列キーによって結び付けられる句がある場合にはそれぞれの句の始点と終点を返す．並列キーに連接する並列句が存在しない（並列キーが並列の役割を果たしていない）場合には NONE を返す．以下はタスクの入出力の例である．

[*1] 文中の複数の並列キーを区別するため，本稿では並列キーを”$word_{index}$”の形式で表す

入力 ”But they also are to see that taxpayers get all allowable tax benefits and to ask if filers who sought IRS aid were satisfied with it.”

出力 but_1: NONE, and_{14}: (5, 13), (15, 26)

3. 提案手法

本研究では並列キーに対して個々の並列句の範囲ではなく並列構造全体としての境界を同定する．単語数 n の文 $x = \{x_1, x_2, x_3, \ldots, x_n\}$ における並列キー x_k に対して，並列構造の前部を $s_1 = \{x_i, \ldots, x_{k-1}\}$ $(1 \leq i \leq k-1)$，後部を $s_2 = \{x_{k+1}, \ldots, x_j\}$ $(k+1 \leq j \leq n)$ とし，それらの可能な組み合わせのそれぞれにスコア計算を行い，最もスコアの高い組み合わせを並列構造の範囲として定める．解析時には同定した並列構造全体の範囲から個別の並列句の範囲を復元する．**図 3** は本研究で用いるニューラルネットワークのアーキテクチャの概要である．提案モデルは以下の 4 つの部分で構成される．

入力層: 単語・品詞の one-hot ベクトルからなる系列に実数値のベクトルを割り当てる．

中間層: 双方向型リカレントニューラルネットワークにより，単語・品詞のベクトルの系列から文脈情報を考慮した隠れ状態のベクトルの系列を取り出す．

特徴抽出関数: 並列構造の範囲の可能な組み合わせについて，双方向型リカレントニューラルネットワークの出力を用いて特徴ベクトルを抽出する．

出力層: 個々の並列構造の範囲の候補に対して，特徴ベクトルからスコア計算を行う．

以降の小節ではこれらのネットワーク構造の詳細について説明をする．

3.1 パラメータ行列による単語・品詞ベクトルの割り当て

本研究で提案するニューラルネットワークのモデルの入力として，語彙数次元の one-hot ベクトルとして表現された単語・品詞の系列 $\{x_t^{word}\}_{t=1}^{T}$，$\{x_t^{tag}\}_{t=1}^{T}$ を受け取る．これらの単語・品詞の one-hot ベクトルの系列はそれぞれ行列 E^{word} $(\mathbb{R}^{v_{word} \times d_{word}})$, E^{tag} $(\mathbb{R}^{v_{tag} \times d_{tag}})$ から，単語・

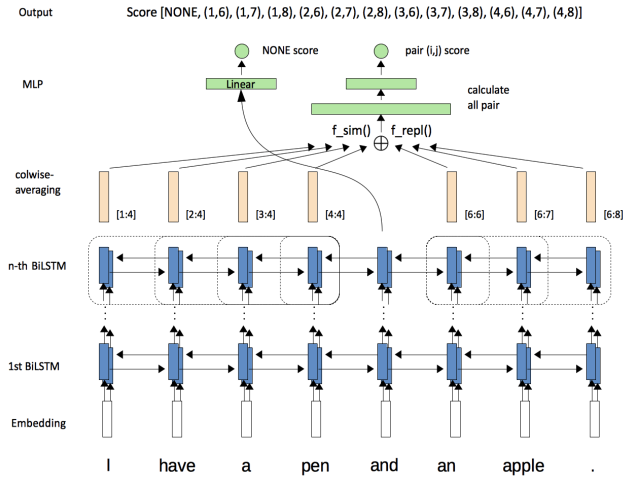


図 3: 並列構造解析のニューラルネットワークのアーキテクチャ

品詞ベクトル $h_{0,t}^{word} \in \mathbb{R}^d$, $h_{0,t}^{tag} \in \mathbb{R}^d$ が割り当てられ，連結される．

$$\begin{aligned} h_{0,t}^{word} &= W^{word} x_t^{word} \\ h_{0,t}^{tag} &= W^{tag} x_t^{tag} \\ h_{0,t} &= [h_{0,t}^{word}; h_{0,t}^{tag}] \\ h_0 &= \{h_{0,1}, \ldots, h_{0,n}\} \end{aligned} \tag{1}$$

3.2 双方向型リカレントニューラルネットワークによる文の表現

入力層にて得られたベクトルの系列を双方向型多層リカレントニューラルネットワークに入力することで出力の系列を計算する．第 n 層の時刻 t における順方向の隠れ状態ベクトル $h_{n,t}^f$ は, 同じ層の直前の時刻 $t-1$ における隠れ状態ベクトル $h_{n,t-1}^f$ と直前の層の同一時刻 t における隠れ状態ベクトル $h_{n-1,t}$ から計算される．

$$h_{n,t}^f = f(h_{n,t-1}^f, h_{n-1,t}) \tag{2}$$

第 n 層の時刻 t における逆方向の隠れ状態ベクトル $h_{n,t}^b$ も同様に計算される．本研究で用いる双方向型多層リカレントニューラルネットワークは層ごとの出力で順方向の隠れ状態ベクトルの系列 $\{h_{N,t}^f\}_{t=1}^T$ と逆方向の隠れ状態ベクトルの系列 $\{h_{N,t}^b\}_{t=1}^T$ が各時刻 t で連結される．一般的なリカレントニューラルネットワークでは関数 $f(\cdot)$ は以下のように定義される．

$$f(x_t, h_{t-1}) = g(Wx_t + Uh_{t-1})$$

関数 $g(\cdot)$ は双曲線正接関数 $tanh(\cdot)$ や正規化線形関数 $ReLU(\cdot)$ などの任意の非線形関数を用いる．リカレントニューラルネットワークは勾配の消失の問題から学習が困難であるため [12]，本研究ではリカレントニューラルネットワークの関数 $f(\cdot)$ に代わって LSTM（Long Short-Term Memory）[6] を用いる．

3.3 特徴抽出関数による並列構造の特徴ベクトル化

双方向型リカレントニューラルネットワークの出力の系列を $h = \{h_t\}_{t=1}^T$ とする．並列構造の前部・後部の候補の特徴ベクトルは系列 h と関数 $g(\cdot)$ から計算される．本研究では関数 $g(\cdot)$ に要素ごとの平均をとる関数を用いて，並列構造の前部・後部のベクトル表現 v_i^{pre}，v_j^{post} を求める．

$$\begin{aligned} g(h_{l:m}) &= \text{average}(h_l, h_{l+1}, \ldots, h_{m-1}, h_m) \\ v_i^{pre} &= g(h_{i:k-1}) \ \ (1 \leq i \leq k-1) \\ v_j^{post} &= g(h_{k+1:j}) \ \ (k+1 \leq j \leq n) \end{aligned} \tag{3}$$

関数 $g(\cdot)$ により得られたベクトル v_i^{pre}，v_j^{post} とベクトルの系列 $h = \{h_t\}_{t=1}^T$ から並列句の類似性・可換性を考慮した特徴ベクトルを計算する．

類似性に基づく特徴ベクトル

類似性に基づく特徴ベクトルは以下のように定義をする．

$$f_{sim}(v_i^{pre}, v_j^{post}) = [|v_i^{pre} - v_j^{post}|; v_i^{pre} \odot v_j^{post}] \tag{4}$$

ここで $|v_i^{pre} - v_j^{post}|$ はベクトルの要素ごとの差の絶対値であり，$v_i^{pre} \odot v_j^{post}$ はベクトルの要素ごとの積である．これらの差・積は類似度を表す尺度として見ることができる [7], [5].

可換性に基づく特徴ベクトル

可換性に基づく特徴ベクトルは以下のように定義をする．

$$\begin{aligned} f_{repl}(h_i, h_j) = [&|h_{i-1} \odot h_i - h_{i-1} \odot h_{k+1}|; \\ &|h_j \odot h_{j+1} - h_{k-1} \odot h_{j+1}|] \end{aligned} \tag{5}$$

$|h_{i-1} \odot h_i - h_{i-1} \odot h_{k+1}|$ は並列構造の直前の文脈と並列前部・後部の接続の差分について計算しており，$|h_j \odot h_{j+1} - h_{k-1} \odot h_{j+1}|$ は並列構造の直後の文脈と並列前部・後部の接続の差分について計算している．ただし $i = 0$ や $j = n$ の場合はそれぞれゼロベクトルとなる．

3.4 フィードフォワードニューラルネットワークによる並列範囲のスコア計算

並列構造の範囲に対して類似性・可換性に基づく特徴ベクトルをそれぞれ抽出し，フィードフォワードニューラルネットワークによってスコアを計算する．

$$\begin{aligned} &\text{Score}(i, j) = \\ &\quad \text{MLP}([f_{sim}(v_i^{pre}, v_j^{post}); f_{repl}(h_i, h_j)]) \end{aligned} \tag{6}$$

また，並列キーに対して並列構造が存在しない場合についてのスコアを計算するため，Score(NONE) を並列キーに対する隠れ状態ベクトル h_k を線形関数に入力することでスコアを求める．

$$\text{Score}(\text{NONE}) = v \cdot h_k + b \tag{7}$$

以上のスコア関数を用いて，長さ n の文の k 番目の単語

に現れる並列キーに対して，並列構造の可能な範囲の組み合わせ (i, j) と並列構造の非存在（NONE）について全てのスコアを計算する．候補は $(k-1) \times (n-k)+1$ 個取り出され，スコアの最も高い候補を並列構造の範囲として決定する．

3.5 モデルの学習

ニューラルネットワークのパラメータ集合 θ は各並列キーに対して正解となる並列構造の範囲との交差エントロピー誤差の和に L2 正則化項を加えたものを確率的勾配法で最小化する．

$$E(\theta) = -\sum_{d=1}^{D}\sum_{i} l_i \log y_i + \frac{\lambda}{2}\|\theta\|^2 \qquad (8)$$

D は学習サンプル数，l_i は正解範囲の分布である．

3.6 デコーディング

本研究のアルゴリズムは並列構造の前部を構成する個々の並列句については範囲を同定していない．3 つ以上の並列句が存在する場合（並列構造の前部に 2 つ以上の句が含まれる場合）は並列構造の前部から個々の並列句を取り出すことで全ての並列句について範囲が同定される．本研究が取り組む手法をベースとした個々の並列句の範囲同定は，並列構造の始点・終点の同定結果に依存するため，本稿では個々の並列句を取り出すアルゴリズムについては取り扱わない．本研究が対象としている英語の並列構造解析において，比較実験のための性能の評価時には並列構造前部に現れるカンマを並列句間の句切れとして個々の並列句に分割する．

4. 関連研究

既存の研究では並列句の類似度に基づく手法が提案されてきた．黒橋ら [11] は日本語の並列構造解析において並列句の類似度計算用の表を用いて動的計画法によって並列構造の検出と範囲の同定を行った．新保ら [14] は英語の並列構造解析において系列アラインメントと単語・品詞・形態情報に基づく素性により並列句の内部の複数単語間の類似度を動的計画法を用いて計算した．黒橋らの手法において表の経路に付与されるスコアはあらかじめ定義された少数のルールに基づくスコア関数によって付与されていた．これに対して新保らの手法では，編集グラフの枝と頂点に与えられるスコアは人手で設計された素性の重み付き線形和で表され，重みのパラメータの調整は機械学習の手法であるパーセプトロンが用いられた．新保らのモデルは入れ子となる並列構造を扱うことができなかったが，原ら [4] は新保らの手法を拡張し，複数の並列構造・句を導出するためのルールを設けることで，スコアの総和が最も高くなるような木に基づいて複数の並列構造・句の範囲を同定した．

表 1: コーパスにおける並列キーの出現数
（括弧内は対応する並列構造が存在する場合のみの集計）

	出現数	文数
Penn Treebank	27903 (24450)	21314 (19095)
訓練データ	22670 (17893)	17282 (13932)
開発データ	953 (848)	742 (673)
評価データ	1282 (1099)	985 (873)
GENIA コーパス	3598 (3598)	2508 (2508)

類似度に基づく並列構造解析の手法に対して，Ficler ら [2] は並列句の類似度に加えて並列句の可換性についても並列句候補のスコア計算の素性として取り入れた．Ficler らの手法は 3 つのコンポーネントから成り立っており，並列構造の検出のための 2 値分類器，並列句の範囲の候補を抽出するための外部の構文解析器（Berkeley Parser[13]），並列句候補のスコア計算によって範囲を一意に決定する識別器という構成となっている．並列句候補のスコア計算においては，人手で設計した素性ではなくニューラルネットワークを用いることで素性設計のコストの問題を克服している．類似度計算には構文情報を用いて並列句をベクトル表現しており，ベクトル同士のユークリッド距離を素性として用いることでグラフを用いた類似度計算手法に比べて計算量が削減されている．可換性については LSTM を用いて並列句の接続部分の文脈情報を考慮した素性ベクトルを使用している．Ficler らの手法は GENIA コーパスにおいて原らの類似度に基づく手法を上回る精度を達成しているが，素性ベクトルの計算において構文情報に依存しているため，3 つのコンポーネントの誤り伝搬や構文解析に派生する誤りや実行時間の点で課題が残る．

河原ら [8] は類似度の素性を用いずに依存構造と格フレームに基づいて並列句の生成確率を学習し，範囲の同定を行っている．吉本ら [18] はグラフベースの依存構造解析の手法を拡張し，依存構造解析とともに並列構造の範囲を同定するアルゴリズムを提案している．

5. 実験

本研究では提案手法の評価実験を並列構造のアノテーションが付与された Penn Treebank コーパス [1] と GENIA コーパス（beta）[9] で行う．各コーパスにおける評価対象となる並列キーの出現数と文数を**表 1** にまとめる[*2]．

5.1 Penn Treebank での実験

5.1.1 実験設定

並列構造のアノテーションがされた Penn Treebank の Wall Street Journal パートのうち，セクション 2 から 21 を訓練データ，セクション 22 を開発データ，セクション 23 を

[*2] 先行研究との比較のため，Penn Treebank の並列キーは”and”, ”or”, ”but”, ”nor”, ”and/or”とし，GENIA コーパスの並列キーは”and”, ”or”, ”but”とする．

評価データとした．単語ベクトルは English Gigaword コーパス第 5 版の New York Times パートを Word2Vec[*3]のデフォルトのパラメータで事前に学習した 200 次元のベクトルを用いた．品詞については区間 $[-1, 1]$ の一様分布でランダムに初期化された 50 次元のベクトルを用いた．また品詞は Stanford Parser[16] を用いて訓練データに対する 10 分割ジャックナイフ法にて付与した品詞を使用した．双方向型リカレントニューラルネットワークには 3 層の双方向型 LSTM を用いて，各単方向の LSTM の隠れ状態のベクトルの次元数は 600 次元とした．フィードフォワードニューラルネットワークは隠れ層を 1 層とし，活性化関数には ReLU（Rectified Linear Unit）を用い，ユニット数は 2400 とした．モデルのパラメータはバッチサイズ 20 のミニバッチを利用した確率的勾配降下法（SGD）で最適化を行った．学習率は Adam[10] を用いて自動調整した．訓練時には Embedding の出力，双方向型リカレントニューラルネットワークの中間層，フィードフォワードニューラルネットワークの隠れ層に対して Dropout[15]（ratio=0.5）を適用し，正則化項のハイパーパラメータ λ は 0.0001 とした．学習のエポック数は 50 に設定し，開発データの F 値（whole）が最も高いエポックで評価データでの評価を行った．

ハイパーパラメータの選択

ハイパーパラメータの設定は，下記の選択肢の中から，開発データの精度が最大となるような設定値を選択している．

- リカレントニューラルネットワークの隠れ状態のベクトルの次元数: {400, 600}
- フィードフォワードニューラルネットワークのユニット数: {1200, 2400}
- ドロップアウト率: {0.33, 0.50}
- 正則化項の λ: {0.0001, 0.0005, 0.001}

5.1.2 評価指標

Ficler ら [2] と同様に，並列キーの前後の並列句の一致（inner）について適合率と再現率の調和平均である F 値によって評価を行う．また本研究では合わせて並列構造の始点と終点での一致の評価（whole），並列構造の一番最初と最後の並列句の一致の評価（outer），全ての並列句の一致の評価（exact）についても行う．また，それぞれの評価方法において，全ての並列句に加えて名詞句の並列構造について評価を行った結果についても示す．Ficler らと同様に NP に加えて NX の並列構造も名詞句の並列構造と見なす．なお本研究の提案モデルが学習・予測する並列構造の範囲は，並列構造の始点と終点（whole）に相当し，inner，outer，exact の評価は 3.6 節で述べている方法で並列構造を個々の並列句に分割して行っている．

[*3] https://code.google.com/archive/p/word2vec/

表 2: Penn Treebank での評価（全並列構造）

	開発			評価		
	P	R	F	P	R	F
Berkeley	70.14	70.72	70.42	68.52	69.33	68.92
Zpar	72.21	72.72	72.46	68.24	69.42	68.82
Ficler et al.	72.34	72.25	72.29	72.81	72.61	72.7
Ours	74.07	71.10	**72.56**	73.46	72.16	**72.81**

表 3: Penn Treebank での評価（名詞句（NP）並列）

	開発			評価		
	P	R	F	P	R	F
Berkeley	67.53	70.93	69.18	69.51	72.61	71.02
Zpar	69.14	72.31	70.68	69.81	72.92	71.33
Ficler et al.	75.17	74.82	74.99	76.91	75.31	**76.1**
Ours	77.43	74.59	**75.99**	75.87	74.76	75.31

表 4: 並列構造の評価方法の違いによる結果
（Penn Treebank 開発データ）

	全並列構造			名詞句（NP）並列		
	P	R	F	P	R	F
whole	75.92	72.87	74.36	77.90	75.05	76.45
outer	72.48	69.57	70.99	76.24	73.45	74.82
inner	74.07	71.10	72.56	77.43	74.59	75.99
exact	72.11	69.22	70.63	75.77	72.99	74.35

5.1.3 実験結果

表 2 に本研究と先行研究の実験結果の比較を示す．本研究での提案モデルは現時点で報告されている Ficler らの state-of-the-art の結果を F 値で 0.11 上回った．名詞句の並列構造についての評価結果は**表 3** のとおりである．本研究で提案するモデルは並列構造の始点・終点について学習・予測しており，個々の並列句については簡易なルールに基づいて分割している．したがって提案手法の学習・予測の性能について示すため，**表 4** に並列構造の始点・終点での評価結果を示す．また，個々の並列句の一致についても評価方法ごとに性能評価をした．

whole の指標において適合率と比較して再現率が低いことから，提案手法では並列キーに対して誤って並列構造が存在しないという予測をしていると考えられる．並列構造内の個々の並列句の一致の評価については，並列キー前後の並列句ではなく並列構造の先頭と最後の並列句を同定するほうが，並列構造解析のサブタスクとして適当である．

5.2 GENIA コーパスでの実験

5.2.1 実験設定

原ら [4] の先行研究との比較のため，GENIA コーパス（beta）における提案手法の性能を評価する．実験設定は前

表 5: GENIA コーパスでの評価（再現率）

種別	#	Ours	Ficler et al.	Hara et al.
全体	3598	**65.98**	64.14	61.5
NP	2317	**66.59**	65.08	64.2
VP	465	63.87	**71.82**	54.2
ADJP	321	78.50	74.76	**80.4**
S	188	**52.65**	17.02	22.9
PP	167	53.89	56.28	**59.9**
UCP	60	50.00	**51.66**	36.7
SBAR	56	78.57	**91.07**	51.8
ADVP	21	**85.71**	80.95	85.7
Others	3	33.33	33.33	**66.7**

述の 5.1.1 における設定値をベースとする．単語ベクトルには BioASQ[17] が提供している 200 次元のベクトルを利用した．この単語ベクトルは学術文献の検索サイトである PubMed[*4]から利用できる生物医学系の論文アブストラクトから Word2Vec を使用して学習したものである．品詞タグは原らと同様に gold の品詞を用いた．正則化項の正則化項のハイパーパラメータ λ は 0.0005，学習のエポック数を 20 とした．実験の評価は 5 分割交差検定によって行った．

5.2.2 評価指標

原らの実験における評価方法と同様に，並列キーに対応づけられる並列構造全体での始点と終点の一致（whole）について，再現率によって評価を行う[*5]．本研究の手法では並列構造の始点と終点を学習・予測するモデルを提案していて直接比較が可能なため，個々の並列句のデコーディングおよび評価は行わない．

5.2.3 実験結果

交差検定を行ったコーパス全体での評価を**表 5** に示す．並列構造の始点と終点の一致の評価について，提案手法は Ficler ら [2] と原らの手法による再現率を上回った．また，並列構造の句構造の種別による結果においては従来手法と比べて文（S）の並列において再現率が顕著に上回った．並列句の類似性のみに着目する原らの手法では捉えられなかった文の並列について改善されたことが分かる．動詞句（VP）や従属節（SBAR）においても原らの手法よりも高いスコアとなっているが，Ficler らの手法より 10 以上低いスコアとなった．名詞句（NP）については Ficler らより高いスコアを達成しているものの，Penn Treebank の実験においては Ficler らの手法は本研究の提案手法より精度が高い．Ficler らの Penn Treebank における実験と原らの GENIA コーパスにおける実験とでは評価指標が異なっているため実験結果を直接比較することはできないが，表 4 の名詞句の並列構造の whole での結果から，本研究の提案手法においても名詞句の並列構造の解析は全並列構造での解析と比較して高精度で行うことができていると分かる．

[*4] https://www.ncbi.nlm.nih.gov/pubmed/
[*5] GENIA コーパスにおいて並列キー”and”, ”or”, ”but”の全てに対して並列構造が存在する．

6. おわりに

本研究では並列構造の範囲同定のために，双方向型リカレントニューラルネットワークを用いて文脈情報を考慮した並列句のベクトル表現を使った手法を提案した．並列句の類似性・可換性の性質に着目し，並列句のベクトル表現と文脈情報から特徴ベクトルを抽出することで，並列句の組み合わせとして最もスコアの高い候補を範囲として決定した．実験により，類似性のみに基づく手法と比べて動詞句や文の並列構造を高い精度で解析できることが示された．提案手法では並列構造内の個々の並列句の範囲は同定していないため，今後は 3 つ以上の並列句に関して解析が行えるようモデルの改善を行う．

参考文献

[1] Ficler, J.: Coordination Annotation Extension in the Penn Tree Bank, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pp. 834–842 (2016).
[2] Ficler, J. and Goldberg, Y.: A neural network for coordination boundary prediction, *arXiv preprint arXiv:1610.03946* (2016).
[3] Hanamoto, A.: Coordination Structure Analysis using Dual Decomposition, pp. 430–438 (2012).
[4] Hara, K., Shimbo, M., Okuma, H. and Matsumoto, Y.: Coordinate Structure Analysis with Global Structural Constraints and Alignment-Based Local Features, Vol. 1, No. August, pp. 967–975 (2009).
[5] Hashimoto, K., Xiong, C., Tsuruoka, Y. and Socher, R.: A joint many-task model: Growing a neural network for multiple NLP tasks, *arXiv preprint arXiv:1611.01587* (2016).
[6] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).
[7] Ji, Y. and Eisenstein, J.: Discriminative Improvements to Distributional Sentence Similarity., *EMNLP*, pp. 891–896 (2013).
[8] Kawahara, D. and Kurohashi, S.: Coordination disambiguation without any similarities, *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, pp. 425–432 (2008).
[9] Kim, J.-D., Ohta, T., Tateisi, Y., Jun’ichiTsujii：GENIA corpus—a semantically annotated corpus for bio-textmining, *Bioinformatics*, Vol. 19, No. suppl 1, pp. i180–i182 (2003).
[10] Kingma, D. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
[11] Kurohashi, S. and Nagao, M.: A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures, *Computational Linguistics*, Vol. 20, No. 4, pp. 507–534 (1994).
[12] Pascanu, R., Mikolov, T. and Bengio, Y.: On the dif-

ficulty of training recurrent neural networks, *International Conference on Machine Learning*, pp. 1310–1318 (2013).

[13] Petrov, S., Barrett, L., Thibaux, R. and Klein, D.: Learning accurate, compact, and interpretable tree annotation, *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 433–440 (2006).

[14] Shimbo, M. and Hara, K.: A Discriminative Learning Model for Coordinate Conjunctions, No. June, pp. 610–619 (2007).

[15] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting., *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958 (2014).

[16] Toutanova, K., Klein, D., Manning, C. D. and Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network, *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, Association for Computational Linguistics, pp. 173–180 (2003).

[17] Tsatsaronis, G., Schroeder, M., Paliouras, G., Almirantis, Y., Androutsopoulos, I., Gaussier, E., Gallinari, P., Artieres, T., Alvers, M. R., Zschunke, M. et al.: BioASQ: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering., *AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text* (2012).

[18] Yoshimoto, A., Hara, K., Shimbo, M. and Matsumoto, Y.: Coordination-aware Dependency Parsing (Preliminary Report), *IWPT 2015*, p. 66 (2015).