

バイパス付き編集グラフを用いた日本語並列構造解析

大熊 秀治[†] 新保 仁[†] 原 一夫[†] 松本 裕治[†]

[†] 奈良先端科学技術大学院大学

[†]{hideharu-o,shimbo,kazuo-h,matsu}@is.naist.jp

あらまし 本稿では、機械学習を用いた日本語並列構造解析手法を提案する。英語においては並列範囲の同定に機械学習を用いた手法が提案されている。英語文において、並列構造の存在は“and”などの手がかり語により容易に検出できるが、日本語並列構造には、英語ほど明確な手がかり表現は存在しないため、並列構造を含むか否かの判定とその範囲の解析を同時に行う手法が必要である。この問題を解決するために、英語既存手法が用いた編集グラフに「バイパス」と呼ぶ非並列文を表すための経路を追加する。また日本語並列構造解析に有効な素性についても検証する。本研究で提案する素性は、シソーラスなどの外部知識に基づく素性や、距離による素性分解の方法、大域的な素性である。EDR コーパスでの実験において、バイパス導入と提案する素性の有効性を確認した。

A Discriminative Learning Method for Japanese Coordinate Phrases Based on Sequence Alignment

Hideharu Okuma[†] Masashi Shimbo[†] Kazuo Hara[†] Yuji Matsumoto[†]

[†]Nara Institute of Science and Technology (NAIST)

Abstract We propose a Japanese coordination structure analysis method based on a discriminative learning technique. For English, an edit-graph-based method exists that effectively identifies the scope of coordinations. Unlike English in which coordinations can be readily detected by cue expressions such as “and,” a variety of cue expressions exist in Japanese and some of them may appear in non-coordinated phrases. Thus, detecting coordinations is also a problem in Japanese coordination structure analysis. To resolve this problem, we introduce a “bypass” in the edit graph to represent a sentence not containing coordinations, so that the feature weights for coordinations are separated from those for non-coordination sentences. In an experiment with EDR corpus, the proposed method outperforms the original Hara’s method.

1 はじめに

並列構造とは、名詞句や動詞句が並列的に出現する構造である。並列構造は、構文解析誤りの原因となる曖昧性を生じさせる。また、並列構造はコーパス中に3, 4割ほど出現するため、その解析誤りは無視できない。

並列構造解析の際に問題となる曖昧性は二つある。一つは範囲の曖昧性で、二つ目は手がかり表現の曖昧性であり、後者は特に日本語で問題となる。

1. 温暖化の(抑制)と(経済)の成長 (誤り)
2. (温暖化の抑制)と(経済の成長)

上の例文は並列構造の範囲の曖昧性の例である。例

文1と2は同じ文だが、助詞「と」を中心とする並列構造の範囲に違いがある。この表現を正解の2番ではなく誤って1番のように解析してしまうと、「温暖化」が「抑制」と「経済」の両方を修飾していると解釈されてしまうため、2番目と意味解釈が異なってしまう。

1. 友達 と¹ 清水寺へ行った。
2. (二条城)と²(清水寺)へ行った。

上記の例文は、手がかり表現の曖昧性の例である。英語においては、並列表現の有無は“and”, “but”などの少数の限られた表現で容易に検出できる。しかし日本語では、上記の例のように手がかり表現が必ずしも並列構造を導かないことがある。また、

並列関係にあるのが動詞の場合は、そもそも明確な手がかり表現が存在しない場合すらある。

並列構造の曖昧性を解消するためのアプローチとして、従来から類似度に基づく手法が広く採用されてきた。日本語では黒橋らが、人手で定義した類似度スコアや手がかり表現を用いる手法を提案している[黒橋 92]。英語では Resnik[Resnik 99] がシソーラスに基づく類似度を用いる手法や、Chantree ら[Chantree 05] がコーパスに基づく分布類似度を用いる手法を提案しているが、任意の並列構造の解析が可能で、機械学習を利用した手法が原ら[原 07]によって提案されている。原らは、並列構造の範囲の同定を同一文内の類似性の解析として定義した。系列アラインメントに基づいて類似性を解析し、各編集操作のスコアは素性の重み付き線形和で定義し、各素性の重みは機械学習の手法で調整した。原らはこの手法を英語の名詞並列構造の解析に適用し、既存の構文解析器よりも高い解析性能を達成した。

日本語並列構造解析は、現在黒橋らの手法が広く採用されている。解析ルールの再整備や、統計的構文解析手法との統合などにより性能が改善されているが、黒橋らの手法自体はそのまま使われている。形態素解析や構文解析の分野においては、統計的機械学習アルゴリズムを用いた手法がルールベースを上回る解析性能を達成しているが、日本語並列構造解析に、機械学習アルゴリズムを適用した研究はまだない。そこで本稿では、原らの手法を改良し、日本語並列構造解析に適用した結果を報告する。

2 系列アラインメントを用いた並列構造解析

2.1 編集グラフによる並列構造の表現

原らのモデルの基礎となるものは、上三角形型の編集グラフ(図 1)である。対角方向の枝は、枝の上と右に位置する語が並列関係にあることを表し、水平、垂直方向の枝は対応する語が削除あるいは挿入されていることを表す。対角方向の枝に、枝によって対応付けられている語の類似度を反映したスコアが与えられていると仮定すると、並列構造の範囲の解析は、編集グラフ内における最大スコアの経路を探索する問題として解くことができる。この時、編集グラフ上の枝は、並列構造の内部(並

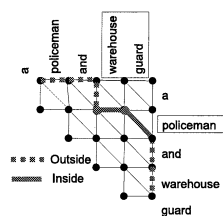


図 1: 英語並列構造解析における編集グラフ

列範囲に対応する部分経路) 可否かで、*Inside* と *Outside* のラベルを付与する。このモデルの詳細に関しては、原らの論文[原 07]を参照してほしい。

類似度スコア関数は幅広い素性の重み付き線形和として定義される。一つの枝のみではなく接続する枝に、周辺単語の情報に基づく素性ベクトルが与えられており、経路の上の全ての素性ベクトルの和を経路に与えられる全体素性ベクトルとすると、経路全体の類似度スコアは重みベクトルと全体素性ベクトルの内積で計算することができる。各素性の重みは、正解経路と推定経路の全体素性ベクトルの差分を取ることで、Perceptron[Collins 02]により学習することができる。

2.2 原らの手法の問題点

原らの手法には三つの問題がある。一つは、素性の開発が十分に行われていないことである。多くの並列構造の解析の研究では、シソーラスや大規模コーパスから計算した統計量などにより解析性能を改善しているが、原らはそれらの情報を用いた素性を使用していないため、系列アラインメントに基づく手法における外部知識に基づく素性の有効性が検証されていない。4 節で、外部知識に基づく素性を提案する。

二つ目の問題点は、並列構造を含まない文の扱いを考慮していないことで、これは日本語に適用する際に特に問題となると考えられる。英語並列構造は、“and” や “or” などの単語によって容易に検出できるため、配列構造の範囲の推定が主なタスクとなる。しかし、1 節で述べたように、日本語のような並列構造の検出が容易でない言語においては、単語情報のみでは検出できない。系列アラインメントの枠組みで、並列構造の有無の判定と範囲の同定を同時に行えるかは自明ではない。3.2 節で、この疑問について議論する。

三つ目の問題点は、大域的素性を利用できないことである。原らは、枝や接続する枝に依存する

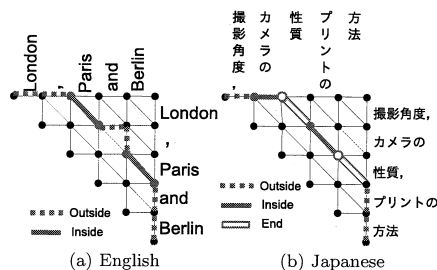


図 2: (a) 英語, (b) 日本語における 3 つの句からなる並列構造。(a) では 2 つのパスは分離しているが, (b) では 1 つになっている。

素性を用いて, Viterbi アルゴリズムによる効率的な探索を行っている。性能をさらに向上させるための方法として, 並列構造の前後の単語や, 並列構造に含まれる単語数などの, 並列構造範囲が決定してから値が決まる「大域的な」素性の使用が考えられる。しかし, 大域的な素性が存在すると, Viterbi アルゴリズムによる探索が不可能になってしまい, 効率的に最適解を求めることができない。この欠点の対応策については, 3.3 節で提案する。

3 日本語並列構造解析のための改良

3.1 IOE モデル

原らのモデルは, 三つあるいは四つ以上の並列句からなる構造を扱うために, 複数の並列句を連続する二つの並列句のペアに分解した。例をあげると, “(A),(B),&(C)” という並列構造は, (A,B) と (B,C) という二つのペアに分解される。グラフ上で部分経路として表されるときは, これらのペアは鎖状につながった部分経路として表される。英語において三つの並列句がある場合は図 2(a) のようになる。図からわかるように, カンマや接続詞 “and” が並列句の間にあるため, 並列構造を表す部分経路が二つあることが認識できる。

日本語並列構造においては, *Inside* と *Outside* の二つのラベルでは, (A,B) と (B,C) での二つの並列構造を認識できない。日本語構文解析は, 周辺情報が豊富に扱えることから文節単位で処理するのが一般的である。並列構造の手がかりとなる助詞の “と” や句読点は機能語であり, 文節内に包含されている。そのため二つの並列構造を表す部分経路はつながってしまい, 各並列構造の範囲が

不確かになってしまう。図 2(b) は日本語で並列句が三つある場合の例である。二つの並列構造を表す部分経路がつながっていることがわかる。

このような不確定性を解消するために, 我々は新しいラベル, *End* を追加して, 並列構造を表す部分経路の最後の枝を他の枝と区別した。従って, 用いられるラベルは全部で三つ, *Inside*, *End*, *Outside* となる。我々はこのラベルを用いるモデルを “IOE モデル” と呼ぶ。図 2(b) において白抜きで表されている枝が *End* ラベルの枝であり, 部分経路の終端を表している。

3.2 バイパス付き編集グラフ

英語においては, 並列構造を含む文は “and” や “or” などの単語で抽出することができる。したがって, 英語並列構造を解析する場合は, “and” や “or” といった手がかり語を含む文のみを対象に並列構造を解析すれば良い。原らの手法は, “and” を含む, 並列構造を含むことが既知の文のみを対象に, 並列構造の範囲を同定する。日本語においては, 並列構造を含む文は 「と」 や 「や」 などのキーとなる表現を含むことが多い。しかし, 1 節で述べたように, 日本語の手がかり表現には, 並列構造を含まない文でも出現するものがある。また, 述語が並列になっている場合には, そもそも明確な手がかり表現が存在しないことすらある。したがって日本語においては, 並列構造が文内に存在するか否かの判定, 存在する場合はその範囲の同定, という 2 種類の解析を行う手法が必要である。

原らの手法でも, 並列構造を含まない文を表現することは可能である。図 4(a) のように, 編集グラフの右上端を通り, 全ての枝が並列構造外部を表す *Outside* からなる経路が, 並列構造を含まない文に対応する。しかし, *Outside* の枝は, 並列構造を含まない文を表現する以外に, 並列構造を含む文においても, 複数の *Inside*/*End* 部分経路をつなぐ, という役割も兼ねている (図 4 の中央の破線)。つまり, 1 個のラベル *Outside* が複数の役割を担っており, これが並列構造の有無の判定に悪影響を及ぼしていると考えられる。そこで, 本研究では図 3(b) のようなバイパス付きの編集グラフを提案する。まず, 編集グラフの右上端のノードを削除する。原らの手法では, *Outside* の垂直方向から水平方向への遷移を禁止し, *Outside* の対角方向の枝を定義していないため, この修正により編

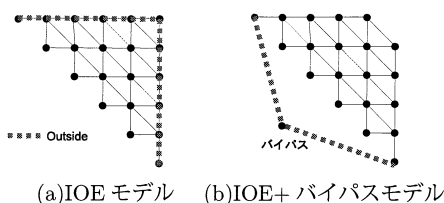


図 3: BIOE モデルとバイパスによる非並列文の表現

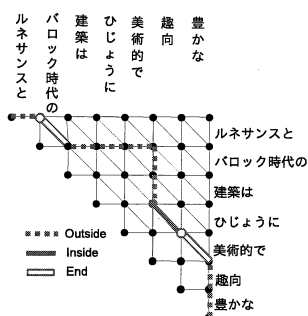


図 4: 並列構造を二つ含む文

集グラフ上の経路は常に並列構造を表す。さらに、編集グラフの始点と終点を直接結ぶ“バイパス”を呼ぶ経路を追加し、バイパスには文全体の特徴を表す素性を用いる。例えば、文の長さ（文節数）や、文中に出現する助詞などが挙げられる。編集グラフの経路の類似度スコアと、バイパスのスコアを比較して、バイパスのスコアが高ければ入力文は並列構造を含まない文と判定して、編集グラフ上の経路の示す並列構造を棄却するという、ある種の閾値処理を行う。元々の編集グラフでは、並列構造を含まない文を並列構造外部の枝の経路で表現していたが、並列構造を含むか否かによって文を分けることで、範囲の解析のための素性と並列構造の有無の検出のための素性を独立に学習させる。

3.3 大域的素性を用いたリランキング

2.2 節で述べたように、原らの手法では、各枝の始点や、連続する枝の接続点に依存する局所的な素性しか用いることができない。しかし、並列構造の推定には、並列構造の範囲が決定してから値が定まる、大域的な素性が有効であると考えられる。下の例文は述語並列の例である。この並列構造は、各並列部の終端文節が「(連用形, 終止形)」というペアで、並列構造の直前に助詞「は」を伴

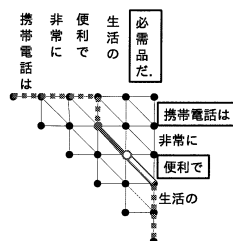


図 5: 大域的素性の例

う文節がある、という特徴を持っている。

携帯電話は (非常に便利で、) (生活の必需品だ。)

この並列構造は、各並列部の終端文節が「(連用形, 終止形)」というペアで、並列構造の直前に助詞「は」を伴う文節がある、という特徴を持っている。この特徴を素性として利用するには、並列構造の直前の文節と、各並列部の終端の文節の情報を同時に見る必要があるため、枝に与えられる素性では表現できない。例えば図 5 において、並列構造の始点の枝を決める時点では、並列構造の終点 (終端文節) がわからないため、並列構造の終端文節が「～だ」という形式であることは分からない。同様に、終点の枝を決める時点では始点の位置がわからないため、並列構造の直前に助詞「は」と伴う文節があることがわからない。したがって、局所的素性では上記の 3 つの位置の属性を同時に素性に使うことができない。

局所的な素性のみを用いる場合は、最適経路は Viterbi アルゴリズムによって文長 n に対して二乗の計算量で効率的に探索できる。しかし、大域的素性を用いると、素性の値を決めるために、並列構造を示す部分経路の始点と終点を決める必要があるため、最適経路を二乗の計算量で探索することができない。

この問題を解決するために、大域的な素性を K-best 解のリランキングのみに用いることで、計算量の増加を抑えて近似的に最適解を探索する。基本アイデアは、素性空間の分割とリランキング処理である。素性空間は、局所的な素性と大域的な素性の二つに分けられる。まず局所的素性を用いて、スコアが高い順に K 個の経路を出力する。これは Viterbi アルゴリズムにより各端点までの最適経路のスコアが計算されていれば、逆向きに A^* アルゴリズムを適用することで効率的に求められる。 K 個の経路が求まったら、経路が示す並列構造の

範囲に基づいて大域的素性の値を決定し、局所的素性のスコアと合わせて最もスコアが高い経路を出力する。

Perceptron 学習時の重みの更新の条件は以下のどちらかが満たされるときである。

1. リランキング後の最適解が不正解
2. リランキング後の最適解は正解だが、局所的素性による最適解 (Viterbi アルゴリズムによる最適経路) が不正解

基本的には通常の perceptron と同様に、リランキング後の最終的な出力が不正解であれば全体の重みを更新する。ただし、二番目の条件を満たすときは、局所的素性の重みのみを更新する。これにより、局所的素性により求まる K 個の解の中に、正しい解が含まれるように重みが調整される。同様の手法が、Kazama ら [Kazama 07] によって固有表現認識のタスクに適用されている。

4 日本語並列構造解析のための素性

本研究で提案する日本語並列構造解析のための素性は、局所的素性と大域的素性に分けられ、その中でさらに内部素性と外部素性に分けられる。局所的な素性は枝や枝の接続に依存する素性で、大域的な素性は並列構造の範囲が決まってから値が決定する素性である。

4.1 局所的素性

4.1.1 内部素性

内部素性は、文節が持つ品詞や助詞などの情報(属性)のみに基づく素性で、シソーラスやコーパスからの統計量といった外部知識を参照しない素性である。内部素性の多くは、基本的には [原 07] で用いられたものに基づくが、日本語に適用するために、単一の属性に基づく素性に加えて、助詞や句読点、述語の活用形などの複数の属性の組み合わせに基づく素性も使用する。これは、日本語並列構造の検出には、助詞や句読点、述語の活用形などの組み合わせが有効と考えられるからである。

4.1.2 外部素性

外部素性とは、シソーラスやコーパスに基づく統計量といった外部知識を利用した素性である。本研究ではシソーラスと共起情報に基づく素性を導

入し、その有効性を検証する。

シソーラス素性 シソーラスとして、分類語彙表 [国立 04] を用いた。分類語彙表では各語に五桁のコード (分類番号) が与えられており、各桁の値はシソーラス木における語の位置 (語が所属する意味的なクラス) を表している。我々はシソーラスの情報に依存する二つの素性を用いる。

一つ目は、二つの単語がシソーラス木のどの深さで同じ意味クラスに属しているかを表す素性である。黒橋らも同様に、シソーラス木において二つの語が一致する位置に依存する素性が用いられている。

二つ目は、分類番号のバイグラムである。シソーラス木の深さに依存する素性は、一致する深さが同じであれば素性は単一の素性として扱われるため、均一な重みが素性に与えられる。しかし分類番号のバイグラムを用いることにより、どの意味クラスの語の組み合わせが並列になりやすいのか、なりにくいのかといった特徴を、各バイグラムごとに独立の重みとして学習することができる。

共起素性 共起表現に関連する素性も利用する。“A の B と C が” というフレーズがあったときに、“A の B”，“A の C” の両方がコーパスに表れやすいならば、並列構造として $((A \text{ の } B \text{ と})(C \text{ が}))$ という構造よりも $(A \text{ の } ((B \text{ と})(C \text{ が})))$ という構造を他に手がかりがなければ選ぶべきである。この特徴を素性として組み込むためにまず共起表現を収集する。毎日新聞 10 年分を KNP で係り受け解析処理を行ったデータを利用した。コーパスに出現する全ての接続関係、依存関係にある文節の自立語のペアを集め、Dunning による尤度比検定の手法 [Dunning 93] を用いて抽出される自立語のペアを共起表現とする。

局所的な素性としては、接続の共起表現のみを利用し、編集グラフで並列構造の外部と内部の境界となる点で用いる。係り受けコロケーションは、6.3 節で説明する大域的素性で用いる。

4.2 文節間の距離による素性分解

本節では新しい素性を導入するのではなく、これまでで説明した局所素性を編集グラフで出現する位置によって二つに分ける新しい基準を提案する。簡潔に言うと、各素性は、編集グラフで素性が出現する位置 (つまり素性が与えられる枝の始点)

と編集グラフの対角線との距離が閾値 θ 以下であるかに依存する素性と、距離に依存しない素性の二つに分解される。この距離は、素性が与えられる枝に対応する文節間の距離を表している。これにより、元々は一つだった素性が二つの素性に分解され、それぞれ独立の重みが学習される。ある素性が条件 X に基づく素性だったとすると、その素性は次の二つの素性、

$$f_1 = \begin{cases} 1 & X \text{ が成り立ち、かつ対角線との距離が } \theta \text{ 以下、} \\ 0 & \text{それ以外} \end{cases}$$

および

$$f_2 = \begin{cases} 1 & X \text{ が成り立つ場合} \\ 0 & \text{それ以外} \end{cases}$$

に分解される。素性を分解することでそれぞれに独立な重みが与えられるので、並列関係にある文節は近い距離にあるという傾向を異なる重みとして学習することができる。5 節の実験では $\theta = 5$ とする。

4.3 リランキングのための大域的素性

リランキング時に用いる大域的な素性は、並列構造周囲の文節の属性の組み合わせ、係り受けコロケーション、並列構造の範囲に関する制約の三つである。このうち、並列構造周囲の属性の組み合わせは内部素性で、係り受けコロケーションは外部素性となる。

並列構造周囲の文節の属性の組み合わせ 並列構造の境界において、境界周囲の文節の助詞や句読点、活用形などの組み合わせを素性として用いるが、並列構造周囲の文節の属性値の組み合わせも、並列構造の境界を定めるのに有効であると考えられる。具体的には、並列構造前後の文節、並列構造の直前の文節と各並列部の終端文節、並列構造の前後の文節と並列構造後部の終端文節の位置の属性を利用する。

係り受けの共起 局所的素性と同様に共起情報に基づく素性を用いる。並列構造の直前文節の自立語は、並列構造前部と後部の終端文節と係り受け関係になりやすいと考えられる。例えば、「社長は(広い邸宅を持ち、)(多くの使用人を雇っている。)」というような文であれば、「社長は」は並列前部「広い邸宅を持ち、」と後部「多くの使用人を雇っている。」の両方の終端文節の「持つ」と「雇う」に係

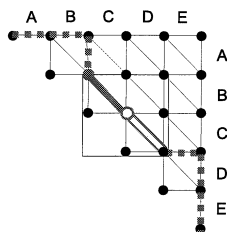


図 6: 並列部がオーバーラップする例

る。このような特徴を素性として表現するために、並列構造直前の文節と、並列構造各部の終端文節の自立語がそれぞれのペアが収集された係り受けの共起表現にあるかどうかを素性として用いる。

並列構造の範囲に関する制約 原らの手法では、局所的な素性のみに基づく Viterbi アルゴリズムにより最適経路を探索するため、並列構造としてはありえない経路を出力することがある。図 6 は並列句がオーバーラップしてしまう例で、並列構造を構成する部分経路を箱型に展開したときに、箱の左下隅が編集グラフの対角線をはみ出してしまっている。このような経路は、全体の経路が定まってからでなければ排除できないため、枝に割り当てられる局所的な素性ではなく、経路全体の素性、つまり大域的な素性を用いなければ排除できない。

最適経路が対角線をはみ出してしまう場合でも、2 番目、3 番目と対角線をはみ出さない経路を探すことで、大域的な素性を用いなくても正常な並列構造を持つ解は出力可能である。しかし、2 番目、3 番目の解が正常な経路でも、先頭も終端も間違った並列構造を示す経路である可能性がある。したがって、Viterbi アルゴリズムによる経路が並列構造として異常でも、並列範囲の一部(先頭あるいは終端)が正解している場合は、そのまま排除せずに出力した方が望ましい。そこで、対角線をはみ出す経路は排除するのではなく、経路がはみ出すか否かに基づく素性の重みによってペナルティを与えてスコアを減点し、減点後のスコアが正常な経路よりも高ければ排除せずにそのまま出力する。

5 実験

本実験では、バイパスにより並列構造を含む否かで文を区別することの効果と、前節で提案した各素性の有効性を検証することを目的とする。

5.1 実験設定

実験には EDR 電子化辞書に収録されたコーパス (EDR コーパス)[情報 95]の一部を用いる。EDR コーパスは約 20 万文からなるコーパスで、新聞記事や雑誌、百科事典、科学辞典などから構成されている。単語単位で品詞情報や意味関係のタグが付与されており、文の構造は S 式による木構造で表される。本実験では、並列構造が含まれる文の割合が大きい平凡社百科事典のデータを用いる。

EDR には単語単位で意味的な関係を表すタグが付与されており、並列関係は“and”によって表される。“and”が付与された単語ペアから構文木を上位にたどっていくと、並列構造の範囲を単語単位で獲得することができる。この時、並列構造が入れ子構造になっている並列構造は、提案手法では扱えないため除外する。また、並列構造を含んでも、構文木上で単語の入れ替えが起きている文も除外する。これは、構文木を上位にたどって取得できる並列構造の範囲が、単語の入れ替えにより連続的にならない可能性があるためである。この処理によって、並列構造を含む文がコーパスから 3,754 文抽出される。

提案手法は、形態素情報と文節区切りの情報を入力とする。本研究では、KNP と性能を比較するため、形態素情報は JUMAN(version 5.1)¹、文節区切りは構文解析システム KNP(version 2.0)²の結果を用いる。また、上述の処理で取得した単語単位の並列範囲も、KNP の文節区切りに合わせて文節単位に変換する。基本的には、単語単位の並列範囲を完全に含む最小の文節列を文節単位の並列範囲とする。ただし、一文節内に並列範囲が含まれてしまう文、並列範囲内の単語が二つの文節にまたがる文、タグ付けの間違いと思われるものは除外した。この結果、並列構造を含む文は 3,257 文となる。これと、並列構造を含まない文 4,192 文を合わせた 7,449 文を実験に用いる。

5.2 評価基準

本実験では、評価基準として次の二つを用いる。

- 範囲の一致
並列構造を構成する各並列部の範囲が正解構造の示す範囲と一致していれば正解
- 終端文節の一致

表 1: バイパスの有効性

モデル	精度	再現率	F 値
バイパスあり (提案手法)	55.0	57.6	56.3
バイパスなし (原らの手法)	55.3	52.1	53.6

表 2: 各素性の有効性

素性	精度	再現率	F 値
内部素性	54.0	57.0	55.4
+ 外部素性	55.0	57.6	56.3
+ 外部素性 + 大域的素性	55.3	57.7	56.5

各並列部の終端文節が正解構造の示す各終端文節と一致していれば正解

KNP は各並列部の終端文節を、並列関係を表す“P”というタグで関係づけるため、終端文節の一致は KNP と比較するための基準である。

性能は精度、再現率およびその調和平均の F 値で評価する。少数の簡単な並列構造のみを出力すれば精度が上がるが再現率が下がる。逆に並列構造を過剰に出力すれば再現率は上がるが精度は低下する。したがって、どちらか一方の値で性能を比較しても正当な評価はできず、最終的な性能の比較は二つの値の調和平均である F 値で評価するのが妥当である。

各実験の結果は、分割交差検定を行った結果で、分割数は 5 である。

5.3 結果

表 1 に、バイパスありとなしのモデルの解析結果を示す。大域的素性は利用していない。バイパス導入により精度は少し下がるが、再現率が大幅に改善されている。バイパス導入により原らの手法よりも F 値で 2.7 ポイント高い性能を達成することができた。表 2 に、各素性の有効性を示す。内部素性に外部素性、大域的素性を加えていくと性能が改善されていることが確認できる。表 3 に、KNP と全ての素性を用いたバイパスモデルの性能を示す。KNP は京都テキストコーパス [黒橋 97] のタグ付け基準に合わせてルールが調整されている [黒橋 00]。EDR コーパスと京都テキストコーパスでは並列のタグ付け基準が異なる可能性があるため、あくまで参考ではあるが、全ての評価指標において、提案手法が KNP より上回っていることがわかる。機械学習の手法により、学習データさえあれば、ルールの調整にかかるコストを大幅に削減しながら、高い解析性能を達成することができた。

¹<http://nlp.kuee.kyoto-u.ac.jp/nl-resource/>

²<http://nlp.kuee.kyoto-u.ac.jp/nl-resource/>

表 3: KNP との比較 (参考)

モデル	精度	再現率	F 値
KNP	58.8	65.3	61.2
バイパスモデル (内部素性 + 外部素性 + 大域的素性)	67.0	69.9	68.4

6 おわりに

バイパス付き編集グラフを提案し、原らの手法よりも高い解析性能を達成することができた。本実験で用いた EDR コーパスと、KNP のルールの調整に用いられた京都テキストコーパスでは並列のタグ付け基準が異なる可能性があるためあくまで参考ではあるが、機械学習の手法により、ルールベースの手法を上回る解析性能を達成することができた。ルールベースの手法は、タグ付け基準が異なると再び人手によるルール調整が必要になるが、その調整にかかる人や時間のコストは大きい。しかし、機械学習に基づく手法では、学習データさえ用意すれば適応は容易である。

リランキング処理により大域的素性を導入したが、本研究では単純な素性を用いた。その結果、範囲の一致について 0.2 ポイントの性能の改善が見られたが、大域的素性の使用はまだ実験的な段階であるため、改良の余地が残されている。今後の展望の一つとして、任意の素性の重みを調整できる機械学習のメリットを活用し、より効果的な大域的な素性の開発を行うことが挙げられる。

また外部知識の有効性を確認することができたが、外部知識の使用についても改善の余地があるだろう。「製品」と「サービス」というように直感的に似ている単語ペアの類似性が分類語彙表では認められなかった。また、「高速で大容量の～」という並列構造であれば、そもそも「大容量」という単語は分類語彙表に登録されていない。しかし「高速」と「大容量」の二つの単語でウェブを検索してみると、二つの単語が並列的に並んだ表現が数多く見られる。したがって、分類語彙表が並列構造の解析に適しているとは限らない。ウェブなどの大規模なコーパスが利用できる現在、機械学習による大規模な類義語情報などの獲得をすべきと考える。

参考文献

- [Chantree 05] Chantree, F., Kilgarrieff, A., de Roeck, A., and Willis, A.: Disambiguating coordinations using word distribution information, in *Proc. Int'l Conference on Recent Advances in Natural Language Processing*, Borovets, Bulgaria (2005)
- [Collins 02] Collins, M.: Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms, in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)* (2002)
- [Dunning 93] Dunning, T.: Accurate methods for the statistics of surprise and coincidence, *Computational Linguistics*, Vol. 19, pp. 61–74 (1993)
- [Kazama 07] Kazama, J. and Torisawa, K.: A New Perceptron Algorithm for Sequence Labeling with Non-local Features, in *Proc. EMNLP/CoNLL*, pp. 315–324 (2007)
- [Resnik 99] Resnik, P.: Semantic similarity in a taxonomy, *J. Artificial Intelligence Research*, Vol. 11, pp. 95–130 (1999)
- [原 07] 原 一夫, 新保 仁, 松本 裕治: アラインメントと機械学習を応用した並列句解析, 人工知能学会論文誌, Vol. 22, No. 3, pp. 248–255 (2007)
- [国立 04] 国立国語研究所: 分類語彙表, 大日本図書 (2004)
- [黒橋 92] 黒橋 禎夫, 長尾 真: 長い日本語における並列構造の推定, 情報処理学会論文誌, Vol. 33, No. 8, pp. 1022–1031 (1992)
- [黒橋 97] 黒橋 禎夫, 長尾 真: 京都大学テキストコーパス・プロジェクト, 言語処理学会第 3 回年次大会, pp. 115–118 (1997)
- [黒橋 00] 黒橋 禎夫: コーパスが先か, パーサーが先か, 情報処理学会論文誌, Vol. 41, No. 7, pp. 1215–1220 (2000)
- [情報 95] 情報通信研究機構: EDR 電子化辞書, <http://www2.nict.go.jp/r/r312/EDR/index.html> (1995)
- [Chantree 05] Chantree, F., Kilgarrieff, A., de Roeck, A., and Willis, A.: Disambiguating coordinations using word distribution infor-