2008－NL－187 (8)
2008／9／24

GENIA コーパスからのネスト並列句同定

原 一夫　新保 仁　大熊 秀治　松本 裕治
奈良先端科学技術大学院大学 情報科学研究科
〒 630-0192 奈良県生駒市高山町 8916-5
{kazuo-h, shimbo, hideharu-o, matsu}@is.naist.jp

本稿では、ネストした並列句も含めた一般の並列句の範囲同定を行うための手法を提案する。本手法ではまず、並列句の範囲同定に特化した文法を定義し、これに基づいて与えられた文を構文木に変換する。その後、CKY 法とパーセプトロン学習を組み合わせたアルゴリズムで素性の重みを学習する。GENIA コーパスを用いた実験で、この手法により、既存の構文解析器を F 値で 11.4 ポイント上回る精度を得ることに成功した。

A Discriminative Method for Detecting the Scope of Coordinated Phrases from Sentences

Kazuo Hara　Masashi Shimbo　Hideharu Okuma　Yuji Matsumoto
Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma Nara 630-0192 Japan
{kazuo-h, shimbo, hideharu-o, matsu}@is.naist.jp

In this article, we present a discriminative method for detecting the scope of coordinated phrases. The method builds syntactic trees from sentences, using a simple grammar taylored for coordination analysis. Each grammatical derivation is associated with features, and these features are multiplied by the current weight vector to yield the score of the derivation. To optimize the weight vector for the training data, we propose a learning algorithm that combines CKY and perceptron. The method achieves an F-score 11.4 points higher than the baseline parser in an experiment with the GENIA corpus.

1 はじめに

自然言語処理研究において、並列句同定は困難な問題の一つである。たとえば、既存のすぐれた構文解析器（Charniak and Johnson, 2005）[2] を使用しても図 1 のような誤りを起こすことがある。医学/生物学分野の学術論文テキストには、生命科学実験の問題設定および実験結果の記載が多く、これらは並列句を用いて記述されることが多いため、これらのテキストにおいては同様の並列句解析誤りは、特に顕著である（e.g., 新規手法と既存手法の対比）。実際、GENIA コーパス（500 の MEDLINE アブストラクトからなる構文木タグ付きコーパス）[6] では、1 文につき約 1 つの割合で並列句（COOD タグ）が出現するが、構文解析器（Charniak and Johnson, 2005）による並列句同定の精度は 50%程

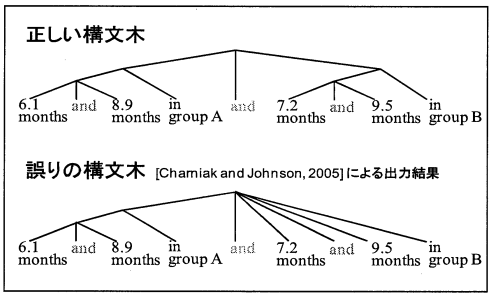

図1: 並列句を含む文は、既存の構文解析器では正しく解析できない場合があることを示す例。
"Median times to progression and median survival times were 6.1 months and 8.9 months in group A and 7.2 months and 9.5 months in group B" の下線部分に対する正しい構文木（上）と、構文解析器 [Charniak and Johnson, 2005] による出力結果（下）。

表1: GENIAコーパスでの並列句出現頻度

並列句の種類	ネストでない並列句	ネストに絡む並列句
名詞句	1797	794
動詞句	312	173
形容詞句	304	105
文	133	150
前置詞句	124	85

度である。

以上の背景のもと、著者らは、並列句の構成要素（conjunct）の類似性を主な手掛かりとして機械学習 (パーセプトロン) に基づく並列句解析手法を開発し、GENIA コーパスを用いた実験で既存の構文解析器を上回る解析精度を得ることに成功した [10]。しかし、この手法には、並列句がネストしている文には適用できないという欠点がある。現実には、表 1 に示すように、医学/生物学分野の学術論文テキストにおいては、並列句は入れ子となって現れることも多く, これらの文を無視することはできない。本稿では、ネスト並列句も含めた一般の並列句の範囲の同定を行うために、CKY とパーセプトロンを組み合わせたアルゴリズムで素性の重みを学習する方法を提案する。

本稿の構成は以下の通りである。まず、今回の提案手法の土台となる逆系列アラインメントを基本とする手法について 2 節で述べる。続いて、並列句を含む文を構文木で表現することによりネスト並列句同定を可能にする方法を 3 節で提案し、4 節で GENIA コーパスを用いた実験結果について報告する。関連研究を 5 節で紹介し、最後に、残された課題について 6 節で述べる。

2 逆系列アラインメントを基本とする手法

著者らは並列句の構成要素（conjunct）が類似しやすいことに着目し、文から並列句の範囲を同定するタスクに（逆）系列アラインメントを適用した [10] 。

一般に、系列アラインメントの目的は、重み値付きの編集操作（e.g., “削除”、“挿入”、“置換”）を用いて系列対を対応させることにより、（編集操作の重み値の合計として）系列対の類似度を測ることである。これとは逆に、系列対に対する編集操作による対応付けを正解として与え、これをうまく再現できるように編集操作（より一般には、素性）の重

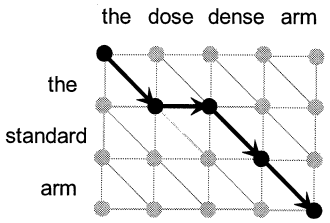

図2: “the standard arm” と “the dose dense arm”の系列対に対する編集グラフと、“置換—挿入—置換—置換”という一連の編集操作に対応する経路

みの値を調整することを目的とするのが、逆系列アラインメントである。

系列アラインメントは、編集グラフ（左上を始点、右下を終点とする有向非循環グラフ）上の経路として表現するのが便利である。たとえば、“the standard arm” と “the dose dense arm” の系列対（並列句の conjunct 対の例である）に対する編集グラフと、“置換—挿入—置換—置換” という一連の編集操作に対応する経路は、図 2 のようになる。ここで、編集操作は、編集グラフ上の節点（格子点と、方向を持った辺の組合せ；図 3 参照）に対応する。

本稿で扱うタスク、すなわち、文から並列句の範囲を同定するタスクにおいては、系列対が同一の文になるため、編集グラフは文をその縦横に配置する上三角形状になる（図 4）。編集操作の種類も、並列句の conjunct としての “並列対応あり” と “並列対応なし” を区別するために、図 3 で示した 3 種類（“並列対応あり”）に加え、“並列対応なし” を意味する 2 種類（“削除”、“挿入”）を追加する必要がある[1] 。

逆系列アラインメントを行うには、まず、素性を定義する。素性は経路の特徴をよく捕えるものが有効であり、素性の例としては、編集操作、編集操作の連接、および、それらと単語との組合せなどである。そして、各素性には重み値を与えておく。次に、経路に対してスコアを計算する。スコアとは、経路上の素性の出現頻度をその重みにより乗じた値を、すべての素性について足しあわせたものとする。

素性の重みは、正解経路が与えられた訓練データ[2] を用いて学習する。学習アルゴリズムはパーセ

[1] 図 4 では、“並列対応あり” の編集操作を黒色、“並列対応なし” の編集操作を灰色で示している。

[2] 実際には、並列句内の経路（＝図 2 のような conjunct の対応付け）が不定であっても、並列句の始点と終点が正解として与えられていればよい [10]。

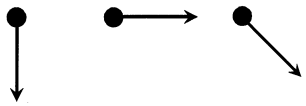

図3: 編集グラフの各格子点上に配置する節点
（左から、“削除”、“挿入”、“置換”）

プトロン [3] による。すなわち、次を繰り返すことで各素性の重み（初期値はゼロとする）を学習する。

(i) 編集グラフ上で（現在の素性の重みのもとで）スコア最大となる経路を動的計画法（ビタビ・アルゴリズム）により算出し、

(ii) (i) で求めた経路が正解経路でない場合に、正解経路とスコア最大経路のそれぞれについて各素性の出現頻度を数え上げ、その差分を現在の重みの値に加えることで各素性の重みを更新する。

3 提案手法

前節の編集グラフ上の経路に基づく手法は、文内に複数の並列句が存在しそれらがネスト構造を持つ場合には、適用することができない。なぜなら、それら並列句に対応する経路が、一つの編集グラフの上で両立できないからである（図 4）。

そこで、本稿では、並列句の範囲同定に特化した文法に基づく構文木によって文を表現し、並列句候補に対して（前節で定義した）系列アラインメントを基本とする素性を割り当て、CKY とパーセプトロンを組み合わせたアルゴリズムで素性の重みを学習する方法を提案する。

ここで用いる文法の目的は、文の全ての構文構造を明らかにすることではなく、あくまで並列句の同定である。このため、句（非終端記号）の種類として、並列句、非並列句しか持たない。さらに、文とその並列句の範囲が与えられたとき、これを一意の構文木に変換できるように、文法を定める。

3.1 並列句の範囲同定に特化した文法

まず、非終端記号と前終端記号を定義する。

- 非終端記号
  - Cc (Coordination; complete)
  - Ci (Coordination; incomplete)
  - N (Non-coordination)

並列句、非並列句に対応する非終端記号をそれぞれ Cc, N とする。Ci は、3 つ以上の構成要素（conjunct）を持つ並列句（e.g., “A, B and C”）を生成する過程で出現する非終端記号とする。

- 前終端記号

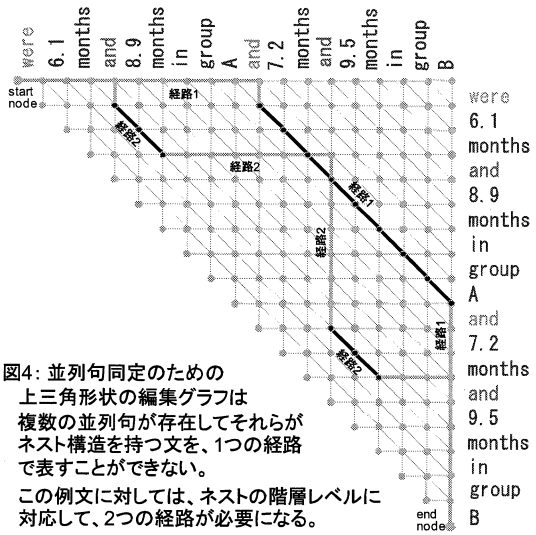

図4: 並列句同定のための
上三角形状の編集グラフは
複数の並列句が存在してそれらが
ネスト構造を持つ文を、1つの経路
で表すことができない。
この例文に対しては、ネストの階層レベルに
対応して、2つの経路が必要になる。

  - CEp (CueExpression; primary ＝主的な並列のキー)
  - CEs (CueExpression; secondary ＝副次的な並列のキー)
  - W (Word ＝並列のキーでない)

ここで、並列のキーとは、並列句の構成要素をつなぐ表現のことを指す。副次的な並列のキーは、Ci と同様、3 つ以上の構成要素を持つ並列句を生成する際に必要となる前終端記号とする。

次に、生成規則について述べる。並列句を左辺に持つ生成規則として、並列句 (Coordination) を 2 つの構成要素 (Left Conjunct, Right Conjunct) と並列のキー (Cue Expression) に書きかえる規則を用意する。すなわち、

- 並列句の生成規則
  - Coordination → LeftConjunct CueExpression RightConjunct

である。具体的には、

- Cc → {N, Cc} CEp {N, Cc}
- Cc → {N, Cc} CEs Ci
- Ci → {N, Cc} CEp {N, Cc}
- Ci → {N, Cc} CEs Ci

とする。ここで、{} は集合であり、その要素のいずれかが、(Left あるいは Right) Conjunct になることを意味する。ネスト並列句は、右辺の conjunct が並列句（Cc）を含む場合に相当する。また、3 つ以上の conjunct を持つ並列句は、最右の 2 つの

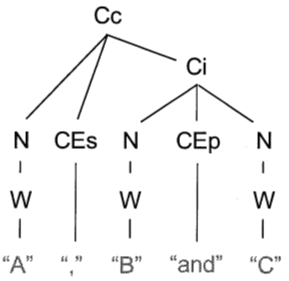

図5: 3つ以上の conjunct を持つ並列句（“A, B and C”）に対する構文木

conjunct でまず不完全並列句（Ci）を作り、残りの conjunct を左から一つずつ付加することで生成する（図 5）。

非並列句を左辺に持つ生成規則は、構文木の冗長性をなくすために便宜的に以下の通りとしておく。

- 非並列句の生成規則
  - N → Cc N
  - N → W Cc
  - N → W N
  - N → W

最後に、前終端記号を左辺に持つ生成規則であるが、これを定めることは、解析対象とする並列句を定めることと同じである。たとえば、次のように定義すると、英語の等位接続詞 “and” に関する並列句のほとんどをカバーすることができる。

- 前終端記号の生成規則
  - CEp → “and”
  - CEp → “, ” “and”
  - CEs → “,”
  - W → (any word)

3.2 並列句候補に素性を割り当てる

上記の文法で生成可能な構文木は、その部分木として並列句（Cc）と非並列句（N）を持ち、それらはそのまま入力文の並列句（候補）の部分とそうでない部分に対応する。他方、文とその並列句の範囲が正解として与えられたとき、上記の文法によりこれを一意の構文木に変換できる。

以上を踏まえ、本稿では、並列句の範囲同定タスクを、上記の文法で生成可能な構文木の中から正解構文木を識別する問題として扱う。識別は、前節に述べた逆系列アラインメントを基本とする方法と同様、訓練データを用いる学習により行う。すなわち、構文木に出現する可能性があるすべての並列句候補に対して素性を割り当て、正しい構文木のスコアが最大になるように素性の重みを学習する。

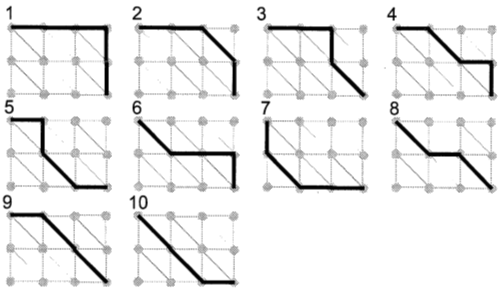

図6: 単語数が、LeftConjunct＝2、RightConjunct＝3の場合、可能な経路の総数は10である。

並列句候補に対して素性を割り当てる具体的方法について、以下に述べる。素性は、並列句を左辺に持つ生成規則に現れる 4 つ組 (Coordination, LeftConjunct, CueExpression, RightConjunct) に対して割り当てる。用いる素性は、前節で用いたもの（編集グラフ上の節点および節点の連接に付与される素性）と同じとする[3]。そして、LeftConjunct と RightConjunct が編集グラフ上に作る長方形領域を考え、その領域内に位置する節点および節点の連接に付与されている素性を、“素性の束” として 4 つ組に割り当てる。

“素性の束” の作り方には、LeftConjunct と RightConjunct は語順を保ちつつ類似しやすいという傾向を反映させる。すなわち、長方形領域内の各節点（あるいは節点の連接）を通過する経路数に比例した割合で、それらに付与されている素性を足し合わせることを行う。たとえば、単語数が LeftConjunct = 2、RightConjunct = 3 の場合、可能な経路[4] の総数は 10 であり（図 6）、各節点に付与されている素性を図 7 に示す割合（通過する経路数÷経路総数）で足し合わせる。長方形の左上から右下への対角線に近いほど割合が高くなることが確認できる。

3.3 CKY とパーセプトロンを組み合わせた学習アルゴリズム

学習アルゴリズムは、前節と同じくパーセプトロンによる。すなわち、次を繰り返すことで各素性の

[3]本稿の提案手法では、編集グラフ上のスコア最大経路探索を行うことなく、素性割り当てのみを目的として編集グラフを用いる。

[4]冗長な経路を除くために、“削除—挿入” の連接は禁止している。

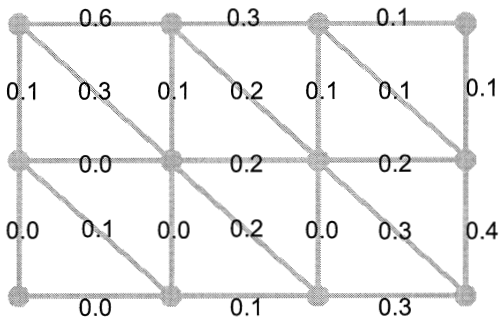

図7: 単語数がLeftConjunct＝2、RightConjunct＝3の場合、各節点（＝格子点と方向を持った辺の組合せ；図3参照）に付与されている素性を、図に示す割合（通過する経路数÷経路総数）で足し合わせることで、"素性の束"を作る。

重み（初期値はゼロとする）を学習する。

(i) 上記の文法で生成可能な構文木の中から（現在の素性の重みのもとで）スコア最大となる構文木を動的計画法（CKY アルゴリズム）により算出し、

(ii) (i) で求めた構文木が正解でない場合に、正解構文木とスコア最大構文木のそれぞれについて各素性の出現頻度を数え上げ、その差分を現在の重みの値に加えることで各素性の重みを更新する。

4 評価実験

GENIA コーパス中の "and" を接続詞として持つ並列句（2,948 個）を解析対象とし、このうちの名詞並列句（1,976 個）の並列範囲を同定するタスクに対して、3 節で述べた本稿の提案手法を適用した。GENIA コーパス（500 文書）を 2 つに分け、一方を訓練データ、他方を評価データとする方法で行った（2 分割交差検定）。

著者らの知る限りでは、並列句の範囲同定に特化した解析器で公開されているものは存在しないため、比較対照は既存の構文解析器（Charniak and Johnson, 2005）[2] とした。構文解析器は句（ブラケット）を出力するが、解析対象の "and" 接続詞に対し、それを含むいちばん内側の句を、その "and" が作る並列句とみなす。構文解析器は句の種類も同時に出力し、"NP" または "NX" を名詞句とする。

名詞並列句範囲同定の結果（適合率、再現率、F 値＝適合率と再現率の調和平均）を、表 2 に示す。構文解析器は GENIA コーパスで訓練していないこと、提案手法はコーパスの POS タグを素性に組み込んでいることを考慮する必要はあるが、提案手法は構文解析器を F 値で 11.4 ポイント上回る精度を得た。

表2: GENIAコーパスを用いた実験結果

手法	適合率	再現率	F値
提案手法	56.8%	54.5%	55.6
Charniak and Johnson, 2005	45.6%	42.9%	44.2

なお、"and" 接続詞が作る句に対して並列句タグ（COOD）が付与されていないものが、わずかであるが存在する。今回は、それらを含む文を解析対象から除外している。また、GENIA コーパスには、並列句（COOD タグが付与されている）が 4,129 個存在するが、"and" を接続詞として持つものが大多数を占める。"and" に続くのは、"or"（約 370 個）, "but"（約 120 個）であるが、これらも解析対象に入れる場合は、3 節で示した前終端記号を左辺に持つ生成規則を変更・追加すればよい。

5 関連研究

本稿での並列句解析の目的は、与えられた英語文に対して名詞並列句をみつけ、その範囲を同定することであった。これに対し、範囲同定の対象を名詞並列句に限定せず、構文解析の精度向上を視野に入れた並列句解析研究（あるいは構文解析の研究そのもの）がある。Kurohashi [7] は、まず並列句の構造（ネストの有無および conjunct の構成）を予測した上で、系列アラインメントに類似した手法を用いて各並列句の範囲を同定している。Charniak and Johnson [2] は、conjunct の類似性も考慮に入れたリランキングを行うことで、構文解析器の精度が向上したことを報告している。

日本語並列句解析では、並列句の係り受け関係を同定することを目的とした研究がある。黒橋と長尾 [11] は conjunct の意味的類似性を基に系列アラインメントの手法を適用し、Kawahara and Kurohashi [5] は conjunct の選択選好を反映した確率的生成モデルを提案している。

一方、並列句を含む名詞句内部の並列構造に関するあいまい性を解消する研究も行われている。"noun1 and noun2 noun3" という形の名詞句に対する次のようなあいまい性

- (noun1 and noun2 noun3)
- (noun1 and noun2) noun3

を、Resnik [9] は単複の一致と大規模コーパスから計算した意味的類似度を、Nakov and Hearst [8] は web から取得した N-gram と paraphrase の頻度を、それぞれ主な素性として用いることで、解消することを試みている。また、Hogan [4] は名詞並列句を対象に、conjunct の構造的類似性と大規模コーパスから計算した head word の意味的類似度を手がかりに Bikel parser [1] の N-best 解をリランキングすることで、あいまい性を解消する方法を提案している。

6 おわりに

本稿では、並列句同定を行うための新手法を提案した。この手法では、並列句解析に特化した文法による構文木で文を表現し、CKY とパーセプトロンを組み合わせたアルゴリズムを用いて素性の重みを学習する。

以下に、今後の展望、課題を述べる。まず、本稿では系列アラインメントを基本とする素性（主に単語 N-gram 単位の素性）のみを用いたが、この手法は本来、並列句単位の特徴を素性に取り入れることができる。たとえば、並列句の大きさ（構成単語数）、並列句の先端と終端の呼応などである。これら素性の有効性を検証することが課題の一つである。もう一つの課題として、本手法の動詞並列句への適用がある。動詞並列句は名詞並列句よりも構成単語数が多く、conjunct の類似性もそれほど顕著でないように思われる。上記の並列句単位の素性を工夫することが、この課題に対する最初の取り掛かりになると考えられる。

また、さらなる課題として本手法の日本語並列句への適用がある。並列のキー表現が “and”, “or” などに限定されている英語とは異なり、日本語は並列のキー表現が多様、もしくはいつも並列句を導くとは限らないことが問題を複雑にすると考えられる。計算量を減らす工夫などが必要になると見込まれる。

参考文献

[1] Bikel, D. M.: A Distributional Analysis of a Lexicalized Statistical Parsing Model, *Proceedings of EMNLP 2004* (Lin, D. and Wu, D.(eds.)), Barcelona, Spain, Association for Computational Linguistics, pp. 182–189 (2004).

[2] Charniak, E. and Johnson, M.: Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking, *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-2005)* (2005).

[3] Collins, M.: Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)* (2002).

[4] Hogan, D.: Coordinate Noun Phrase Disambiguation in a Generative Parsing Model, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, Association for Computational Linguistics, pp. 680–687 (2007).

[5] Kawahara, D. and Kurohashi, S.: Coordination Disambiguation without Any Similarities, *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK, Coling 2008 Organizing Committee, pp. 425–432 (2008).

[6] Kim, J.-D., Ohta, T., Tateisi, Y. and Tsujii, J.: GENIA corpus: a semantically annotated corpus for bio-textmining, *Bioinformatics*, Vol. 19, No. Suppl. 1, pp. i180–i182 (2003).

[7] Kurohashi, S.: Analyzing coordinate structures including punctuation in English, *Proceedings of The Fourth International Workshop on Parsing Technologies*, pp. 136–147 (1995).

[8] Nakov, P. and Hearst, M.: Using the Web as an Implicit Training Set: Application to Structural Ambiguity Resolution, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada, Association for Computational Linguistics, pp. 835–842 (2005).

[9] Resnik, P.: Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language, *Journal of Artificial Intelligence Research*, Vol. 11, pp. 95–130 (1999).

[10] Shimbo, M. and Hara, K.: A Discriminative Learning Model for Coordinate Conjunctions, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 610–619 (2007).

[11] 黒橋禎夫, 長尾眞: 並列構造の検出に基づく長い日本語文の構文解析, 自然言語処理, Vol. 1, No. 1, pp. 35–57 (1994).