

複数の述語項関係を利用した文内ゼロ照応解析

大内 啓樹^{1,a)} 進藤 裕之^{1,b)} Kevin Duh^{1,c)} 松本 裕治^{1,d)}

概要: 述語項構造解析において、省略された項の検出と、それが指示する要素の同定を、特にゼロ照応解析と呼ぶ。従来のゼロ照応解析では、各述語に対して、文内の他の述語項との相互関係を考慮せず、独立に項を決定するというアプローチが主流であった。しかし、文内に出現する述語間には意味的な関連があり、ある述語の項の決定が他の述語の項決定に影響を及ぼすように思われる。したがって、複数の述語項の相互関係に関する情報は、ゼロ照応解析に役立つことが期待される。本稿では、複数の述語項関係を同時に考慮し、決定するモデルを提案し、ゼロ照応解析の精度向上に取り組む。NAIST テキストコーパスを用いた評価実験により、提案モデルの有効性を示す。

1. はじめに

述語項構造解析において、省略された項の検出と、それが指示する要素の同定を、特にゼロ照応解析と呼ぶ。例えば、図1の例文において、「読んでいた。」のガ格の項は省略されている(「 ϕ_i は」)。このような省略された項をゼロ代名詞という。ゼロ代名詞「 ϕ_i は」は「私 $_i$ 」を指す。ゼロ代名詞と、ゼロ代名詞に指される要素(先行詞)間の関係をゼロ照応という。ゼロ照応解析は、ゼロ代名詞を検出し、ゼロ照応関係を同定するタスクである。

述語の項を同定する際、述語との相対的な位置関係によって、各項は次の3つに分類される[4]。

係り受け有 (INTRA-D): 述語と直接係り受け関係にある項。

文内ゼロ (INTRA-Z): ゼロ代名詞の先行詞であり、述語と同一文内に現れる項。^{*1}

文間ゼロ (INTER): ゼロ代名詞の先行詞であり、述語と異なる文に現れる項。

この中で、文内ゼロ (INTRA-Z) と文間ゼロ (INTER) の解析が、ゼロ照応解析に相当する。係り受け有 (INTRA-D) の項の同定が比較的容易である一方、文内ゼロ (INTRA-Z) の解析は困難であり、出現数も多いため、無視できない問題となっている。例えば、図2の例文で、「逮捕したが、」のガ

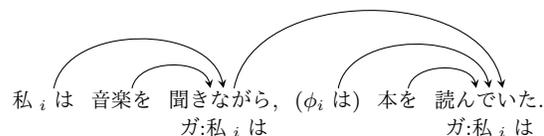


図1 二つの述語が、ガ格の項を共有する例

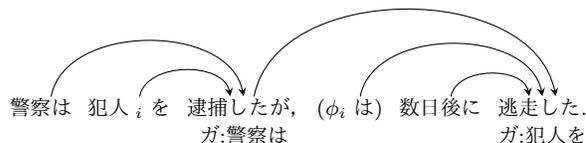


図2 二つの述語が、異なるガ格の項を持つ例

格の項である「警察は」は、直接係り受けがあり、INTRA-Dに相当するため、解析が容易であると推測される。逆に、「逃走した。」のガ格の項は、ゼロ代名詞の先行詞である「犯人 $_i$ を」であり、INTRA-Zに相当するため、解析が困難であると予想される。本研究では、日本語で特に出現頻度が高い、ガ格である文内ゼロ照応 (INTRA-Z) の解析精度向上を目的とする。

従来のゼロ照応解析では、各述語に対して、文内の他の述語項との相互関係を考慮せず、独立に項を決定するというアプローチが主流であった。しかし、文内に出現する述語間には意味的な関連があり、ゼロ照応関係を同定する上で役立つことが期待される。例えば、図1において、「聞きながら、」と「読んでいた。」は「私 $_i$ は」をガ格の項として共有しており、接続助詞(ながら)を伴って述語間に係り受け関係がある。接続助詞(ながら)が、主節と従属節を接続している場合、両節で同一の主語をとることが多い。そのため、「聞きながら、」のガ格の項が「私 $_i$ は」であるならば、「読んでいた。」のガ格の項も同じく「私 $_i$ は」

¹ 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

a) ouchi.hiroki.nt6@is.naist.jp

b) shindo@is.naist.jp

c) kevinduh@is.naist.jp

d) matsu@is.naist.jp

*1 本研究では、先行研究[4][2]と同様に、述語と直接係り受けのない項を INTRA-Z とする。

である可能性が高い。また、図2では、「逮捕したが、」と「逃走した。」の関係を意味的な観点から考えると、同一の動作主がこれらの行動をとる可能性は低いと思われる。そのため、「逮捕したが、」のガ格の項が「警察は」であるならば、「逃走した。」のガ格の項はそれ以外の要素となる可能性が高い。このように、ある述語項は他の述語の項を決定する際に有用な情報となりえる。そこで、本稿では、複数の述語項を同時に考慮し、決定するモデルを提案する。

本稿の主な貢献は次の3つである。

- (1) 複数の述語項を同時に考慮し、決定することを可能とする二部グラフモデルを提案する。
- (2) 乱拓化山登り (Randomized Hill-Climbing) 法 [8] に基づいて、上記の二部グラフモデルの最適解を求める手法を提案する。
- (3) NAIST テキストコーパスを用いた実験により、複数の述語項を考慮しない手法より、提案手法が高い解析精度を実現することを示す (F 値での評価において、全述語項の解析精度が 1.2 ポイント、文内ゼロ照応関係にある項の解析精度が 2.7 ポイント上回る)。

2. 先行研究

日本語述語項構造・ゼロ照応解析で中心的に用いられてきたコーパスの一つに、NAIST テキストコーパス [3] がある。これは、新聞記事と社説から構成される約 40,000 文に、述語項及び照応、共参照のタグを付与したコーパスである。NAIST テキストコーパスにおいて、項の担う意味役割は、ガ格 (主格)、ヲ格 (対象格)、ニ格 (与格) の3種類の表層格で定義されている。本研究でも、NAIST テキストコーパスを利用する。

NAIST テキストコーパスを用いて、ゼロ照応付き述語項構造解析に取り組んだ主な先行研究として、Taira ら [6] と Imamura ら [4] の研究が挙げられる。

Taira らは、SVM 分類器と決定リストを利用した解析手法を提案している。彼らは、動詞・形容詞などの一般的な述語だけでなく、事態性名詞についても項構造解析を行っている。Imamura ら [4] は、大規模コーパスから獲得した言語モデルのスコアなどの文脈的な情報を、対数線形モデルに組み込むことによって、高性能を達成している。

Taira らと Imamura らは、格ごとに別々のモデルを構築している。Sasano ら [5] が指摘しているように、格ごとに有効である素性が異なるため、本研究でも格ごとに独立にモデル化するアプローチを採用する。

また、彼らの手法では、各述語に対して、文内の他の述語項との相互関係を考慮せず、項候補集合から最尤の項候補を選ぶことによって、独立に項を決定する。そのため、同一文内の他の述語の項の決定に影響を及ぼさない。本研究の提案モデルは、先行研究と異なり、複数の述語項関係を素性として組み込み、それらの相互関係を考慮して、文

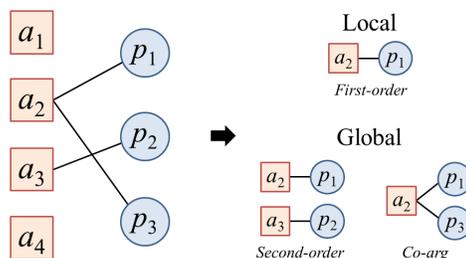


図3 二部グラフモデル

内の全ての述語項を同時に決定する。

3. 大域的素性を利用した二部グラフモデル

3.1 二部グラフモデル

図3は、本稿で提案する二部グラフモデルを表している。右側のノード集合 (述語) に左側のノード集合 (項候補) を割り当てることで、各述語のとる項を表現する。具体的に、文 x に対して、二部グラフ $G(x) = (A_x, P_x, E_x)$ は、項候補集合 A_x 、述語集合 P_x 、エッジ集合 E_x から構成される。これらの各集合は、次の要素から成り立つ。

$$A_x = \{a_1, \dots, a_n, a_{n+1} = \text{NULL}\}$$

$$P_x = \{p_1, \dots, p_m\}$$

$$E_x = \{e_{a_i p_j} \mid \sum_{a_i \in A_x} e_{a_i p_j} = 1, \forall p_j \in P_x\}$$

項候補集合 A_x は項候補 (a_i) から構成され、ダミー項候補 NULL を含む。これは、述語が項をとらない場合や、文外に項がある場合に割り当てられる。述語集合 P_x は述語 (p_j) から構成される。エッジ集合 E_x は、 A_x と P_x 間のエッジ ($e_{a_i p_j}$) から構成され、各エッジは各述語に項が割り当てられることを表す。エッジは、各述語 p_j に対して、 A_x に属するノード a_i から一本のみ引かれる。

あるエッジ集合 E_x を持つ二部グラフ y に対して、重みベクトル θ と高次元素性ベクトル $\phi(x, y)$ の内積によってスコアを定義する。可能な二部グラフ集合 $G(x)$ から、スコア最大の二部グラフ \hat{y} を、次式のスコア関数に従って求める。

$$\hat{y} = \operatorname{argmax}_{y \in G(x)} \theta \cdot \phi(x, y)$$

スコア関数内の重みベクトル θ は、機械学習手法によって推定することができる。本稿では、平均化パーセプトロン [1] を用いて、 θ を推定した。

先行研究では、他の述語の項の決定に影響を及ぼさない素性が、述語と項候補間に定義される。本稿では、これを局所的素性 (First-order 素性) と呼ぶ。それに加え、複数の述語項間の相互関係を表現した大域的素性を新たに定義し、局所的素性と大域的素性の両方を解析に用いる。

| Structure | Name | Description |
|-----------|-------------|---|
| SECOND | PAIR | $\langle a_1.ax \circ a_1.rp \circ p_1.ax \circ p_1.vo \rangle, \langle a_2.ax \circ a_2.rp \circ p_2.ax \circ p_2.vo \rangle$ |
| | TRIANGLE | $\langle a_1.ax \circ a_2.ax \circ a_1.rp \circ a_2.rp \circ p_1.ax \circ p_1.vo \rangle,$ $\langle a_1.ax \circ a_2.ax \circ a_1.rp \circ a_2.rp \circ p_2.ax \circ p_2.vo \rangle$ |
| | QUAD | $\langle a_1.ax \circ a_2.ax \circ p_1.ax \circ p_2.ax \circ a_1.rp \circ a_2.rp \circ p_1.vo \circ p_2.vo \rangle$ |
| CO-ARG | BI PREDS | $\langle p_1.w \circ p_2.w \rangle$ |
| | CO-ARG DEPS | $\langle a_1.ax \circ p_1.ax \circ p_2.ax \circ p_1.vo \circ p_2.vo \circ (x, y).dep \rangle,$ $\langle a_1.ax \circ p_1.ax \circ p_2.ax \circ p_1.vo \circ p_2.vo \circ p_1.t \circ p_2.t \circ (x, y).dep \rangle$ if x depends on y for x, y in $(p_1, p_2), (a_1, p_1), (a_1, p_2), (p_1, a_1), (p_2, a_1)$ |

表 1 大域的素性テンプレート. \circ は各要素の結合を表す; ax=助詞, rp=相対的位置, vo=態, w=基本形, t=品詞, dep=係り受け.

3.2 大域的素性

複数の述語項間の相互関係を考慮した大域的素性を提案する. 大域的素性は, 図 3 における Second-order 構造と Co-arg 構造に基づき, 表 1 に示されている素性テンプレートとして定義する.

Second-order

Second-order 構造は, 二つの述語項から構成される. 二つの述語のうち, 文内で語順が先の述語を p_1 と表し, 後の述語を p_2 と表す. また, p_1 の項は a_1 , p_2 の項は a_2 と表記する. これらの述語項において, 次の 3 つの関係に着目し, 素性テンプレートを定義する.

PAIR : 一組の述語-項の関係.

例) p_1 と a_1 間の関係.

TRIANGLE : 一つの述語と二つの項の関係.

例) p_1, a_1, a_2 間の関係.

QUAD : 二組の述語-項の関係.

例) p_1, p_2, a_1, a_2 間の関係.

各関係において, Second-order 構造における項の相対的な位置関係 (rp:Relative Position) と, 助詞 (ax:Auxiliary), 態 (vo:Voice) の情報を組み合わせて素性とする. 項の相対的な位置関係は, ある項がもう一方の項や二つの述語の前後どちらに位置するかを組み合わせて表現する. 例えば, 項 a_1 の相対的な位置関係を定義する場合, a_2, p_1, p_2 のそれぞれに対して, a_1 が前後どちらに位置するかを調べる. a_1 が a_2 の「後」, p_1 の「後」, p_2 の「前」に位置している場合, それら 3 つを組み合わせ, 「後後前」といった形で表現する.

異なる二つの述語の項を決定する場合, 互いの述語が文内のどの位置の要素を項としているかという情報が, 項決定に役立つと期待し, これらの素性を定義した.

Co-arg

図 1 のように, 文内ゼロ (INTRA_Z) が現れる場合, 複数の述語が同一の要素を項として共有することが多くなる. そのような現象を捉えるため, 二つの述語とそれらが共有する一つの項から構成される Co-arg 構造に基づいて, 素性テンプレートを定義する. 二つの述語のうち, 文内で

語順が先の述語を p_1 , 後の述語を p_2 , 共有された項を a_1 と表す. これらの述語項において, 次の 2 つの関係に着目し, 素性テンプレートを定義する.

BI PREDS : 二つの述語の単語バイグラム.

CO-ARG DEPS : 二つの述語と共有される項の間の係り受け関係.

BI PREDS は, 二つの述語の基本形を組み合わせた素性となっている. この素性は, 二つの述語に対する項の共有しやすさを捉える.

CO-ARG DEPS は, Co-arg 構造を構成する 3 つの要素 (p_1, p_2, a_1) 間の係り受け関係 (dep:Dependency) と, 助詞, 態, 品詞 (t:POS tag) を組み合わせた素性である. 共有される項と述語間や, 二つの述語間の係り受け情報は, ゼロ照応を含む述語項の解析に有用である可能性が高い. 例えば, 図 1 の例では, 接続助詞〈ながら〉を含む文節である「聞きながら」が「読んでいた」に直接係っており, これら二つの述語は「私_iは」を項として共有する. つまり, 前述したように, 接続助詞〈ながら〉で接続された節間では, 同一の主語をとりやすいという性質があるため, 例のように, 直接係り受け関係のある二つの述語間 (主節と従属節間) で, 項を共有するか否かの有力な手がかりとなっている. しかし, 文内に述語が 3 つ以上あれば, 接続助詞〈ながら〉と述語を含む文節が, 直接係らない (接続しない) 述語を含む文節も出現する. その場合, それらの述語が項を共有するとは限らない. したがって, 二つの述語間の係り受け関係や, 共有される項と各述語間の係り受け関係は, それらが伴う助詞と組み合わせることによって, 二つの述語が項を共有するか否かを決定する重要な情報となる.

以上のように, Second-order と Co-arg 構造に基づいて, 大域的素性を定義した. しかし, 大域的素性を用いることにより, スコア最大の二部グラフの探索は困難なものとなる. その解決策として, Zhang ら [8] が提案した, 乱拓化山登り法*2を利用する.

*2 Randomized Hill-Climbing.

```
1: Input: parameter  $\theta$ , sentence  $x$ 
2: Randomly initialize bipartite graph  $y^{(0)}$ ;
3: pred_list = word order predicate list of  $x$ ;
4:  $t = 0$ ;
5: repeat
6:   for each predicate  $p \in \text{pred\_list}$  do
7:      $y^{(t+1)} = \underset{y \in G(y^{(t)}, E_x(p))}{\text{argmax}} \theta \cdot \phi(x, y)$ ;
8:      $t = t + 1$ ;
9:   end for
10: until no change in this iteration
11: return  $\hat{y} = y^{(t)}$ ;
```

図4 二部グラフモデルにおける最適解探索のための乱拓化山登り法

3.3 乱拓化山登り法による探索

乱拓化山登り法は、任意の素性を利用できるという特徴がある。そのため、前節で提案した大域的素性を、自然に扱うことができる。Zhangらは、依存構造解析において、表現力の高い素性ととも乱拓化山登り法を利用し、いくつかの言語で最高精度を達成している[8]。本研究では、乱拓化山登り法に基づいて、我々が提案した二部グラフモデルにおいて、スコア最大の二部グラフを求める手法を提案する。

図4に、二部グラフモデルの最適解探索に適用した乱拓化山登り法の擬似コードを示す。まず、文 x に対して可能な二部グラフ集合 $G(x)$ から、初期二部グラフ $y^{(0)}$ をランダムにサンプリングする(図4の2行目)。次に、文 x における語順通りに並べた述語を格納するリストを用意する(3行目)。 $y^{(t)}$ における当該述語 p のエッジを、他の可能なエッジに替えた場合に得られる二部グラフの中から、最もスコアの高いグラフを選び、 $y^{(t+1)}$ として更新する(6-9行目)。図4の7行目の式において、 $E_x(p)$ は p のとりうるエッジ集合を表す。 $G(y^{(t)}, E_x(p))$ は、 $y^{(t)}$ における p のエッジを、 p のとりうるエッジにそれぞれ替えた際に得られる二部グラフの集合を表す。アルゴリズムは、どの一つの述語のエッジを替えても、スコアが向上しなくなるまで続く(10行目)。結果として、局所最適な二部グラフが得られる(11行目)。

また、Zhangら[8]と同様に、文 x が与えられたとき、このアルゴリズムを K 回繰り返して、得られた K 個の二部グラフの中で、最もスコアの高いものを最終的に解として選ぶ。この K 回の繰り返しの、ランダムリスタートと呼ぶ。ランダムリスタートの回数が多ければ、より良い局所解に辿り着く可能性が高くなるのが期待できる。

4. 評価実験

提案モデルの有効性を調査するため、評価実験を行った。解析対象をガ格の文内項(*INTRA_D*, *INTRA_Z*)に設定し、文外項(*INTER*)は対象外とした。また、評価・解析は、文節単位で行った。

4.1 データセット

我々は、NAIST テキストコーパス 1.5[3]を用いて、ガ格に対する提案手法の評価を行った。実験では、次に示すような、標準的なデータ分割法を採用し、モデルの訓練・開発・評価を行った[6]。

訓練： 1月1-11日の記事と、1月から8月の社説。

開発： 1月12, 13日の記事と、9月の社説。

評価： 1月14-17日の記事と、10月から12月の社説。

実験で用いるモデルの素性として、NAIST テキストコーパスにおいてアノテーションされている品詞タグ、文節境界、係り受け情報を利用した。外部資源は一切利用していない。

4.2 ベースライン

提案モデルと比較するベースラインの手法は、Imamuraら[4]の手法と同様で、一つの述語に対してそれぞれの項候補のスコアを点推定し、各述語に対してスコア最大の項候補を選ぶことによって、述語項構造を求める。素性として、Imamuraら[4]の使用した素性のうち、Additional Featuresとして定義された素性以外を使用する*3。

4.3 実装詳細

解析の際に用いる項候補は、ベースライン・提案手法どちらにおいても、文内に含まれる全文節とした*4。解析対象の述語は、NAIST テキストコーパスでアノテーションされているものを用いた。

提案モデルを利用した解析では、ランダムリスタートを50回($K = 50$)に設定し、解析を行う*5。解析結果として、独立に10回解析した結果の平均を報告する。また、モデルの訓練に関しては、ベースライン・提案手法どちらも、平均化パーセプトロンで訓練した*6。

5. 結果と考察

5.1 大域的素性の効果

表2は、評価データにおけるベースラインのモデルと提案モデルの解析結果を示している。我々の提案モデルは、全文内述語項(*ALL*)、係り受け有(*INTRA_D*)、文内ゼロ(*INTRA_Z*)のそれぞれにおいて、F値で76.68%(+1.25)、85.44%(+1.15)、45.83%(+2.69)の解析性能を達成し、ベースラインの性能を上回った。この結果から、本研究で提案した大域的素性は、文内ゼロ照応解析と述語項構造解析の

*3 本研究では、外部資源を利用せず、文内項のみに着目するため、外部資源や、文外の要素を利用する Additional Features[4]は使用しない。

*4 今後、項候補の効率的な枝刈りの方法なども調査したい。

*5 開発データを用いた予備実験で、解析精度がほぼ収束したのが $K = 50$ のときだったため、評価データにおける解析でも同じ回数とした。

*6 イテレーション数は15に設定した。

| Model | ALL | | | INTRA_D | | | INTRA_Z | | |
|-----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | P | R | F ₁ | P | R | F ₁ | P | R | F ₁ |
| ベースライン 提案モデル | 77.87 78.42 | 73.14 75.02 | 75.43 76.68 | 82.84 83.84 | 85.50 87.09 | 84.29 85.44 | 54.56 55.04 | 35.66 39.26 | 43.14 45.83 |

表 2 評価データを用いた解析結果. ALL=全文内述語項. INTRA_D=係り受け有. INTRA_Z=文内ゼロ. P=精度. R=再現率. F₁=F 値.

| Restart | F ₁ | Sent./Sec. |
|---------|----------------|------------|
| 1 | 76.58 | 862.07 |
| 10 | 76.67 | 240.85 |
| 50 | 76.68 | 68.49 |
| 100 | 76.68 | 34.72 |

表 3 提案手法における各ランダムリスタート回数ごとの解析結果. Restart=リスタート回数. F₁=全文内述語項 (ALL) を対象とした解析性能 (F 値). Sent./Sec.=1 秒間で解析可能な文数.

両方において、有効に機能していることがわかる。

特に、本研究で着目した文内ゼロ照応解析 (INTRA_Z) において、提案モデルがベースラインを F 値で約 2.7 ポイント上回っている。この F 値の向上は、もともと低かった再現率 (R) の大幅な向上 (+3.6 ポイント) が理由となっている。これらの結果から、複数の述語項の関係を考慮する大域的素性は、ゼロ照応関係のカバレッジの向上に貢献していることがわかる。

5.2 ランダムリスタート回数の効果

表 3 は、乱拓化山登り法のランダムリスタート回数を変更して、評価実験を行った結果を示している*7。まず、性能を比較すると、リスタートしなかった場合 (K = 1) の F 値が 76.58% であり、リスタート回数を増やした場合より、約 0.1 ポイント低い。一方、リスタート回数 10 (K = 10) で F 値が 76.67% となり、性能的にはほぼ収束する。予備実験として、リスタート回数を 100 回以上に増やしてみたが、解析精度はほぼ一定であった。

次に、解析速度を比較すると、リスタートしなかった場合 (K = 1) は、一秒間で約 850 文の解析を実現している。当然であるが、リスタート回数を増やすにつれて、解析に時間がかかることがわかる。

これらの結果を踏まえると、初期値とする二部グラフを変更し、リスタート回数を増やしたほうが、より良い局所解 (あるいは大域最適解) に辿り着ける可能性は高くなるが、解析にかかる計算時間は大きくなるというトレードオフがあると言える。今回の実験設定では、リスタートしない場合でも、リスタート回数が多い場合に得られる性能に近い結果が得られた。これは、局所解と最適解の差が小さかったため、少ないリスタート回数で良い性能が得られたと考えられる。また、リスタート回数が少ない方が解析に

*7 提案手法を用いるにあたり、ランダムリスタート回数以外は変更していない。

時間がかからないため、大規模な文書データを高速に処理する必要がある場合などに適していると考えられる。今後、全解探索などで最適解を求め、リスタート回数と最適解到達数の相関に関して詳しく調査する。

5.3 事例分析

提案手法での解析成功事例

図 5 は、提案手法での解析成功事例を示している。「扱い、」と「引き起こす」は、「人たちを」をガ格として共有する。「扱い、」に対する「人たちを」は、直接係り受け関係になく (INTRA_Z)、解析が困難であると予想される。実際、ベースラインの手法では、「扱い、」のガ格は、文外にあると誤って判定されていた。

しかし、提案手法では、Co-arg 構造に基づいて、二つの述語間の関係や、共有される項と、それぞれの述語との関係を考慮し、項の共有しやすさを計算する。この事例の場合、二つの述語間の直接係り受け関係や、「扱い、」の読点手が手かかりとなり、述語の項を独立に決定する手法では正解できなかった事例を、正しく解析できたと考えられる。

提案手法での解析失敗事例

一方、図 6 は、提案手法での解析失敗事例を示している。ベースラインの手法では、「話してくれた。」のガ格は文外に出現すると正しく解析されたが、提案手法では、「人が」がガ格であると誤って解析された。この事例における提案手法は、「いて、」のガ格が「人が」であるなら、「話してくれた。」も同様に、「人が」を項として共有する可能性が高いと判定した。

この事例のように、項を共有しやすい二つの述語があり、かつ、直接係り受け関係などの構文の手かかりが乏しい場合、文外に項をとる述語の解析が困難である傾向が見られた。このような事例の解決策として、文外の項を候補に入れての解析が必要であると考えられる。

6. おわりに

本稿では、述語項構造解析における文内ゼロ照応解析の改善のため、複数の述語項関係を利用した二部グラフモデルを提案した。

評価実験を通して、提案モデルの有効性を調査し、文内ゼロ照応解析の性能向上に寄与することを示した。その理由として、ある述語がゼロ代名詞の先行詞を項とする場合、



図 5 提案手法での解析成功事例

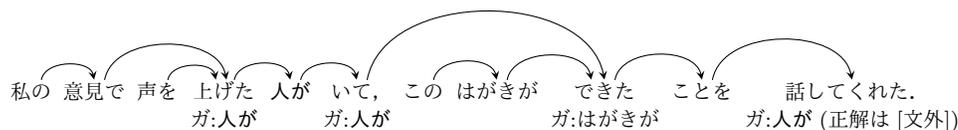


図 6 提案手法での解析失敗事例

同一文内の他の述語も、同じ先行詞を項として共有する場
合が多いため、複数の述語項間の情報を考慮した提案モデ
ルが有効に働いたと考えられる。

複数の述語項関係を同時に考慮することによるゼロ照
応解析の性能向上は、文内項に限定されるものではなく、
文外項にも自然に拡張できると期待される。今後、文外項
(INTER) も含めた解析実験を行っていききたい。

また、乱拓化山登り法のランダムリスタート回数を変更
した実験によって、リスタートが少なくても、解析性能は
大きく損なわれず、高速に解析できることを確認した。こ
れは、局所解が最適解とそれほど差がなかったため、少な
いリスタート回数で良い性能が得られた可能性がある。今
後、リスタート回数と最適解に関する詳細な調査を進めて
いきたい。

参考文献

- [1] Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms, *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, Association for Computational Linguistics, pp. 1-8 (2002).
- [2] Hayashibe, Y., Komachi, M. and Matsumoto, Y.: Japanese Predicate Argument Structure Analysis Exploiting Argument Position and Type., *IJCNLP*, pp. 201-209 (2011).
- [3] Iida, R., Komachi, M., Inui, K. and Matsumoto, Y.: Annotating a Japanese text corpus with predicate-argument and coreference relations, *Proceedings of the Linguistic Annotation Workshop*, Association for Computational Linguistics, pp. 132-139 (2007).
- [4] Imamura, K., Saito, K. and Izumi, T.: Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution, *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Association for Computational Linguistics, pp. 85-88 (2009).
- [5] Sasano, R. and Kurohashi, S.: A Discriminative Approach to Japanese Zero Anaphora Resolution with Large-scale Lexicalized Case Frames., *IJCNLP*, pp. 758-766 (2011).
- [6] Taira, H., Fujita, S. and Nagata, M.: A Japanese predicate argument structure analysis using decision lists, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 523-532 (2008).
- [7] Yoshikawa, K., Asahara, M. and Matsumoto, Y.: Jointly extracting japanese predicate-argument relation with markov logic, *IJCNLP*, pp. 1125-1133 (2011).
- [8] Zhang, Y., Lei, T., Barzilay, R. and Jaakkola, T.: Greed is Good if Randomized: New Inference for Dependency Parsing, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Association for Computational Linguistics, pp. 1013-1024 (2014).