文法・意味的役割に基づく空所化構文の解析

加藤 芳秀　　松原 茂樹

名古屋大学情報連携統括本部
yoshihide@icts.nagoya-u.ac.jp

1 はじめに

空所化 (gapping) とは，等位接続された構成素（等位項）に共通する要素が，片方の構成素から省略される現象である．空所化構文を含む文を意味解析するためには，省略された要素を補う処理が必要となるが，そのような手法についての研究はあまり進んでいない．

本稿では，空所化構文を解析する一手法を提案する．本手法では，空所化を示すタグ，及び文法・意味的役割を表すタグを付与した構文木に基づく構文解析を行い，これらのタグを手がかりに空所化構文を解析する．具体的には，空所化を示すタグが構文解析結果に含まれている場合，その文には空所化構文が含まれているものとして解析する．この解析処理は，等位項に含まれる構成素間の対応付けと位置づけられるが，対応付けにおいては役割を表すタグを活用する．実験により，従来の手法と比べて，本稿で提案するアプローチが大幅に高い再現率を達成できることを確認した．

2 空所化構文の解析

本節では，空所化構文について Penn Treebank (PTB) [5] の表記法をベースに説明し，次に，空所化構文を解析する従来の手法について概観する．

2.1 PTB における空所化構文

空所化とは，等位構造中の等位項に含まれる共通する要素が，片方の等位項において省略される現象である．空所化された等位項において残された要素を**残余要素** (remnant) と呼ぶ．もう一方の等位項は空所化されていないが，この等位項において残余要素と対応関係にある要素を**相関要素** (correlate) と呼ぶ．相関要素を，対応する残余要素で置き換えることにより，空所化されていない等位項が得られる．

PTB においては，残余要素と相関要素の対応関係が与えられている．図 1 に PTB における空所化構文を含む構文木の例を示す．ここにおいて，“-” で番号付けられたラベルを持つノードが相関要素であり，“=” で番号付けられたラベルを持つノードが残余要素である．空所化された等位項はフラットな構造を取っており，すべての残余要素が等位項を表すノードの子ノードとなっている．相関要素と残余要素が同一の番号を持つとき，それらが対応関係にあることを意味している．例えば，この構文木において，NP-SBJ-1 と NP-SBJ=1 はそれぞれ相関要素と残余要素であり，対応関係にある．NP-SBJ-1 を NP-SBJ=1 がルートの構文木で，NP-TMP-2 を PP-TMP=2 がルートの構文木で置き換えると，“the six-month bills will still mature on May 3, 1990” に対する構文木が得られるが，これは，2 番目の等位項において “will still mature” が省略されたことを示している．

2.2 空所化構文の解析に関する従来の研究

本節では，空所化構文の解析に関する従来の研究について概観する．

Ficler らは，PTB の構文木を変換して空所化構文を解析する方法を提案している [2]．この方法では，一方の等位項から相関要素を，もう一方の等位項から残余要素をくくり出し，くくり出されたもの同士を等位接続するような構文木に変換する．この変換は，相関要素，及び残余要素が等位項において右端にまとまって存在することを前提としており[1]，図 1 に示したような構文木には適用できないといった問題がある．

Kummerfeld らは，構文解析結果としてグラフ構造を構成できる構文解析を提案している [4]．この手法の目的は，wh 移動などの痕跡の解析であり，空所化構文を研究のターゲットとしてはいないが，残余要素と相関要素の対応関係を構文構造中のエッジとして処

[1] 具体的には，argument-cluster coordination と呼ばれるタイプの構造にしか適用できない．

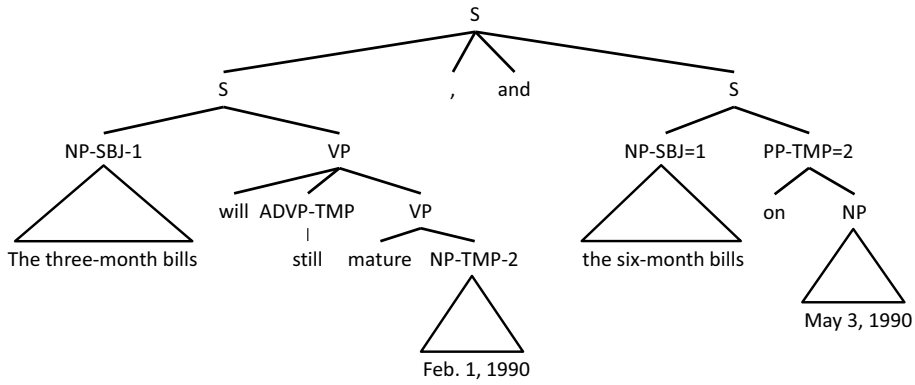

図 1: PTB における空所化構文

理する方法について言及している．しかし，この構文解析手法は，残余要素と相関要素の対応関係に相当するエッジを構文解析結果として生成できない場合が多く，再現率が低いという問題がある．

3　空所化構文の解析

本節では，空所化構文を解析する手法を提案する．空所化構文の解析を実現する上での一つの問題は，空所化構文は出現頻度が低く，その解析モデルを構築するための学習データの量が十分でないことである．この問題を緩和するために，本手法では相関要素と残余要素の対応関係を直接的に学習するのではなく，構文木コーパスに含まれる別の種類の情報を学習し，それを活用して対応関係を同定する．具体的には，提案手法では，以下の 2 種類の情報を構文木上にアノテーションとして埋め込み，アノテーション情報に基づき残余要素と相関要素の対応関係を同定する．

- 等位項が空所化されたか否かを示す情報
- 文法・意味的役割

以下では，まずこれらのアノテーションについて説明し，次に，アノテーションされた情報に基づき空所化構文を解析するための処理について述べる．

3.1　アノテーション

提案手法では，空所化された等位項を構文解析時に同定するために，そのような等位項に対してタグ GAP を付与する．具体的には，PTB の構文木中のノード n が以下の条件を満たすとき，タグ GAP を付与する．

表 1: PTB における文法・意味的機能タグ

文法的		意味的	
DTV	与格	BNF	利益
LGS	論理的主語	DIR	方向
PRD	述語	EXT	程度
PUT	put に関する場所的な補部	LOC	場所
SBJ	表層的な主語	MNR	方法
		PRP	目的
		TMP	時間

- n の子ノードの中に，残余要素，すなわち “=” を用いて番号付けされたノードが存在する．

図 1 においては，範疇 S の構成素が等位接続されているが，右側のノードに対して，タグ GAP が付与される．

次に文法・意味的役割を表すタグについて説明する．一般に，対応関係にある残余要素と相関要素は，文法的，あるいは意味的に同一の役割を担っている．例えば，図 1 中の番号 1 の対応関係において残余要素と相関要素はどちらも主語となっており，番号 2 の対応関係においては，どちらの要素も時間を表している．この事実が示唆するのは，残余要素と相関要素のそれぞれが担う役割の同一性に基づき，対応関係を同定できる可能性である．PTB においては，部分的にではあるが，文法的・意味的役割に関する情報が機能タグとして付与されている．提案手法では，表 3 に示す機能タグを文法・意味的役割を示す情報として用いる[2]．

ここで重要なのは，役割タグを用いる提案手法は，学習データの観点でメリットがある点である．すなわ

[2]文法的な機能タグ PRD を持つノードが，意味的な機能タグも持つことがあるが，この場合，PRD のみを用いて，意味的役割は用いない．

ち，役割タグはPTBに多数含まれているため，これを同定するモデルの構築は容易であると考えられる．

3.2 残余要素と相関要素の対応付け

本節では，前節で述べた方法に従ってアノテーションされた構文木を用いて，残余要素と相関要素を対応付ける方法について説明する．本手法は，次の二つの処理からなる．

- 空所化された等位項を検出し，残余要素と相関要素の候補を取り出す処理
- 残余要素と相関要素の対応関係を同定する処理

以下では，これらを順に説明する．

空所化された等位項（以下，n_{gap}）にはタグ GAP が付与されているため，まず，そのようなノードを見つけることから処理は開始される．n_{gap} が存在するとき，n_{gap} と等位接続された空所化されていない等位項（以下，n_{init}）は，次の条件をすべて満たすノードと定める．

- n_{init} は n_{gap} の左側の兄弟である．
- n_{init} と n_{gap} の範疇は同一である．
- n_{init} の左側の兄弟には，それと同一の範疇をもつノードが存在しない．

このようにして得られた等位項 n_{init} と n_{gap} において，n_{init} に含まれる相関要素と n_{gap} に含まれる残余要素との対応関係を同定するのが後段の処理である．n_{init} の子孫ノードを相関要素の候補とし，n_{gap} の子ノードを残余要素の候補として対応付けを行う．対応付けとして複数の可能性が考えられるが，対応関係の数が最大となるように対応付けを行う．対応関係の数が同数の場合は，相関要素が覆う単語の数の和が多いものを優先する．対応関係にある相関要素 c と残余要素 r は同一の役割を持つという制約を課すが，以下の条件のいずれかが成り立つときその役割は同一であると定義する．

- c と r はともに役割タグを持ち，その役割タグが一致する．
- c と r の範疇はともに前置詞句 PP であり，その主辞となる前置詞が一致する．
- c と r はともに役割タグを持たず，範疇が一致する．

表 2: 精度・再現率

	精度	再現率	F
Kummerfeld ら [4]	100.0	6.9	12.9
提案手法（BERT なし）	66.7	20.7	31.6
提案手法（BERT あり）	88.2	51.7	65.2

対応関係の同定においては，さらに以下のような構造的な制約を課す．なお，対応関係の集合を A とし，相関要素 c と残余要素 r に対応関係があるとき，$(c, r) \in A$ とする．

- $(c, r) \in A$ かつ $(c, r') \in A$ ならば，$r = r'$.
- $(c, r) \in A$ かつ $(c', r) \in A$ ならば，$c = c'$.
- 任意の $(c, r), (c', r') \in A$ について，c が c' の左側に位置するならば，r は r' の左側に位置する．
- 任意の $(c, r), (c', r') \in A (c \neq c')$ について，c と c' はオーバーラップしない．

これらの制約を満たす対応関係は，動的計画法により効率的に求めることができる．

4 評価実験

提案手法の有効性を確認するために，空所化構文の解析に関する性能評価実験を行った．PTB の WSJ コーパスのセクション 02–21, 22, 23 をそれぞれ学習データ，開発データ，テストデータとして使用した．構文解析には，Kitaev ら [3] のものを使用した．この構文解析では，外部データとして BERT[1] を使用できるが，使用する場合と使用しない場合で実験した．学習データと開発データに提案手法によるアノテーションを施し，構文解析モデルを学習した．ハイパーパラメータは，文献 [3] と同一である．テストデータの文に対してこのモデルで構文解析し，解析結果のアノテーションに基づき対応関係を同定した．解析性能の評価は，Kummerfeld ら [4] の評価尺度を用いた．すなわち，対応付けられた残余要素と相関要素について，それらの範疇及び位置を組として，精度・再現率を評価した．

表 2 に精度・再現率を示す．Kummerfeld らの手法では，構文解析結果として残余要素と相関要素の対応関係に相当するエッジを生成することができない場合が多く，再現率が非常に低い．一方，提案手法では，

表 3: タグ同定の精度・再現率

	出現	BERT なし			BERT あり		
タグ	頻度	精度	再現率	F	精度	再現率	F
GAP	16	66.7	25.0	36.4	84.6	68.8	75.9
SBJ	4148	97.4	96.8	97.1	98.1	98.1	98.1
PRD	1025	82.0	78.8	80.4	88.3	85.1	86.6
LGS	166	88.6	88.6	88.6	91.5	90.4	90.9
DTV	19	73.7	73.7	73.7	87.5	73.7	80.0
PUT	10	50.0	40.0	44.4	72.7	80.0	76.2
TMP	1302	89.7	91.9	90.7	91.8	94.3	93.0
LOC	953	88.4	80.9	84.5	91.2	84.5	87.7
DIR	293	71.1	45.4	55.4	82.8	62.5	71.2
PRP	204	76.9	55.4	64.4	84.3	71.1	77.1
MNR	178	76.2	77.5	76.9	72.5	79.8	75.9
EXT	105	87.5	80.0	83.6	88.9	83.8	86.3
BNF	2	0.0	0.0	0.0	0.0	0.0	0.0

精度において劣るものの再現率は大幅に高くなっており，F 値も大幅に向上している．

提案手法においては，構文木に付与されているタグを正しく同定できるかどうかが性能に直結する．そこで，タグ GAP，及び役割タグの同定についての性能を評価した．タグとその出現位置を組とし，精度と再現率を評価した．表 3 に結果を示す．役割タグの同定の精度・再現率は，BERT を使用した場合と使用しない場合とでそれほど大きな差はない．一方，タグ GAP に関しては，BERT を使用しない場合は，精度・再現率がかなり低くなっている．対応関係の同定においても BERT を使用しない場合は，使用する場合に比べて精度・再現率がかなり低くなっており，タグ GAP の同定の性能が，対応関係の同定の性能に大きく影響していると考えられる．

5 おわりに

本稿では，空所化構文を解析する方法として，構文木にアノテーションを行い，その情報に基づき解析する手法を提案した．提案した手法では，構文木に付与された文法・意味的役割に基づき残余要素と相関要素の対応関係を同定する．より解析精度を高めるためには，構成素の文法的・意味的な類似度といった指標を用いて対応付けすることが考えられるが，そのような手法の開発は今後の課題である．また，本稿では構成素構造に基づく空所化構文の解析を扱ったが，依存構造をベースにした空所化構文の解析も提案されており [6]，これと提案手法との比較についても今後，検討したい．

謝辞

本研究は一部，科研費基盤研究 (C)(No. 17K00303) により実施した．

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[2] Jessica Ficler and Yoav Goldberg. Improved parsing for argument-clusters coordination. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 72–76, 2016.

[3] Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2676–2686, 2018.

[4] Jonathan K. Kummerfeld and Dan Klein. Parsing with traces: An $O(n^4)$ algorithm and a structural representation. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 441–454, 2017.

[5] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 310–330, 1993.

[6] Sebastian Schuster, Joakim Nivre, and Christopher D. Manning. Sentences with gapping: Parsing and reconstructing elided predicates. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1156–1168, 2018.