空所化情報を考慮した句構造から依存構造への変換

小林 幹輝 [1]　加藤 芳秀 [2]　松原茂樹 [1,2]
[1] 名古屋大学大学院 情報学研究科
[2] 名古屋大学情報連携推進本部
kobayashi.mikiteru.i1@s.mail.nagoya-u.ac.jp

概要

構文構造の変換は，異なる表現形式に基づく構文解析器間の性能比較に有用である．しかし，その多くは空所化構文を考慮していない．本論文では，空所化構文を考慮した句構造から依存構造への変換手法を提案する．提案手法では，相関要素を残余要素で置き換えることにより，空所化構文において省略された要素の同定，及び省略された要素と残余要素の間の依存関係を生成する．提案手法を用いて，空所化構文を解析する句構造解析器と依存構造解析器の性能比較実験を行い，本変換手法の有効性を確認した．

1　はじめに

構文構造を表現する文法，あるいはアノテーション体系は句構造や依存構造，Combinatory Categorial Grammar（CCG）[1] など様々存在しており，構文解析器の出力結果もそれが基づく文法やアノテーションに応じて異なる．異なる表現形式を採用する構文解析器間の性能を比較する一つの方法として，構文構造を変換するアプローチが考えられる．また，ある文法に基づきアノテーションされたコーパスに対して，変換を施すことにより別のアノテーションのコーパスが容易に得られるという観点からも構文構造の変換は有用である．これまでに様々な構文構造変換手法が提案されている [2, 3, 4, 5, 6, 7]．

代表的な句構造コーパスの一つとして Penn Treebank（PTB）[8] が挙げられる．PTB のアノテーションには句の情報だけでなく，空所化（gapping）[9] の情報も含まれている．空所化とは，等位構造の等位項において，共通する要素が省略される現象である．例えば，“Stock prices closed higher in Stockholm and lower in Zurich.” は，“closed higher in Stockholm” と “lower in Zurich” を等位項とする等位構造であり，後者の等位項は “closed lower in Zurich” から “closed” が省略されたものである．PTB において空所化は，文中における句の対応関係の情報を用いて表現されるが，現在の構文構造変換手法の多くは，この情報を考慮していない．その一因として，PTB に基づく句構造解析器の多くは，空所化の情報を含まない句構造しか出力しないことが挙げられる．しかし近年，空所化の情報を含んだ解析結果を生成する句構造解析器も開発されており，その解析精度も高くなっている [10]．

本論文では，句構造から依存構造への変換において，空所化の情報を考慮した変換手法を提案する．提案手法では，空所化の情報が付与されている PTB の句構造を依存構造へと変換する．変換後の依存構造は，空所化の情報を反映したものとなっている．提案手法の応用例として，句構造解析器と依存構造解析器の性能比較実験を行った．

2　PTB から Enhanced UD への変換

本節では，Penn Treebank（PTB）[8] に基づく句構造を，Enhanced Universal Dependencies（EUD）[11] に基づく依存構造へと変換する従来の手法について概説する．まず，PTB 及び EUD において空所化構文がどのように表現されるかについて説明し，次に変換手法について説明する．

2.1　空所化構文

空所化とは，等位接続された句（**等位項**）に共通する要素が片方の句から省略される現象である．空所化を含む文を空所化構文と呼ぶ．空所化構文において，省略の起きている等位項に残された要素を**残余要素**（remnant）と呼び，もう片方の等位項において残余要素に対応している要素を**相関要素**（correlate）と呼ぶ．空所化構文の例として下記のような文が挙げられる

（1）　Stock prices closed higher in Stockholm and

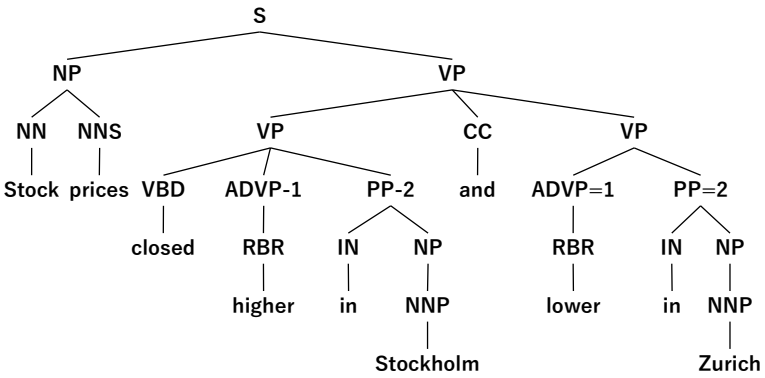

図 1　PTB における空所化構文の例

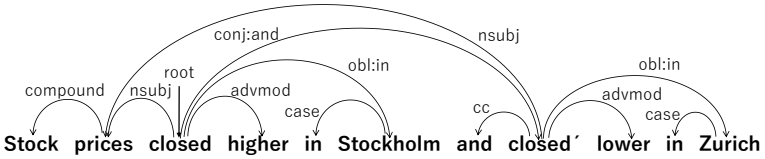

図 2　EUD における空所化構文の例

Ø lower in Zurich.

ここで “Ø” は，要素が省略された位置を示している．“closed higher in Stockholm” と “Ø lower in Zurich” が等位項である．右の等位項は，もともと “closed lower in Zurich” であったものが，動詞 “closed” が等位構造内で共通していることから，それが省略されたものとみなすことができる．文（1）において，“lower” と “in Zurich” が残余要素であり，“higher” と “in Stockholm” がそれぞれに対応する相関要素である．

2.2　PTB における空所化構文

Penn Treebank（PTB）は句構造に関する代表的なコーパスの一つである．PTB に基づく句構造解析器は多数開発されている．PTB のアノテーション規則に従ったコーパスとして GENIA コーパス [12] や English Web Treebank[13] などがある．以下では，本論文において焦点となる空所化構文について例を用いて説明する．

PTB における句構造の例を図 1 に挙げる．PTB のアノテーションでは，空所化の情報も付与されている．“=” で番号付けられたラベルが残余要素を表す．図 1 の構文木では，“ADVP=1”，及び “PP=2” とラベル付けされているノードをルートとする部分木が残余要素である．対応する相関要素は，“-” で番号づけられている．図 1 では，“ADVP-1”，及び “PP-2” とラベル付けされているノードをルートとする部分木が相関要素である．“ADVP=1” と “ADVP-1” が，“PP=2” と “PP-2” がそれぞれ対応関係にある．相関要素を残余要素で置き換えることで，“closed lower in Zurich” に対する構文木が得られる．これは，2 番目の等位項 “lower in Zurich” では “closed” が省略されていることを示している．

2.3　EUD における空所化構文

Enhanced Universal Dependencies（EUD）[11] は，現在広く用いられている依存構造のアノテーションである Universal Dependencies（UD）[14] を拡張したものである．EUD の依存構造は UD の依存構造への新たな依存関係の追加，及び依存関係の詳細化によって得られ，UD に比べてより詳細に構文構造を表現できる．

文（1）に対する EUD の依存構造を図 2 に示す．EUD では，残余要素や相関要素をアノテーションするのではなく，空所化により省略された要素を補って依存構造を構成する．例えば文（1）の場合は，“lower in Zurich” に対して省略された “closed” を補う．これにより，等位項 “lower in Zurich” を “closed lower in Zurich” という省略のない句のように扱うことができるため，“prices” を “closed´” の主語，“lower” を “closed´” の修飾語といった依存構造として捉えられる．

2.4　PTB から EUD への変換

PTB の句構造から EUD の依存構造への変換手法としては，Schuster らの手法 [11] が存在する．この手法では，PTB から EUD への変換をパターンマッ

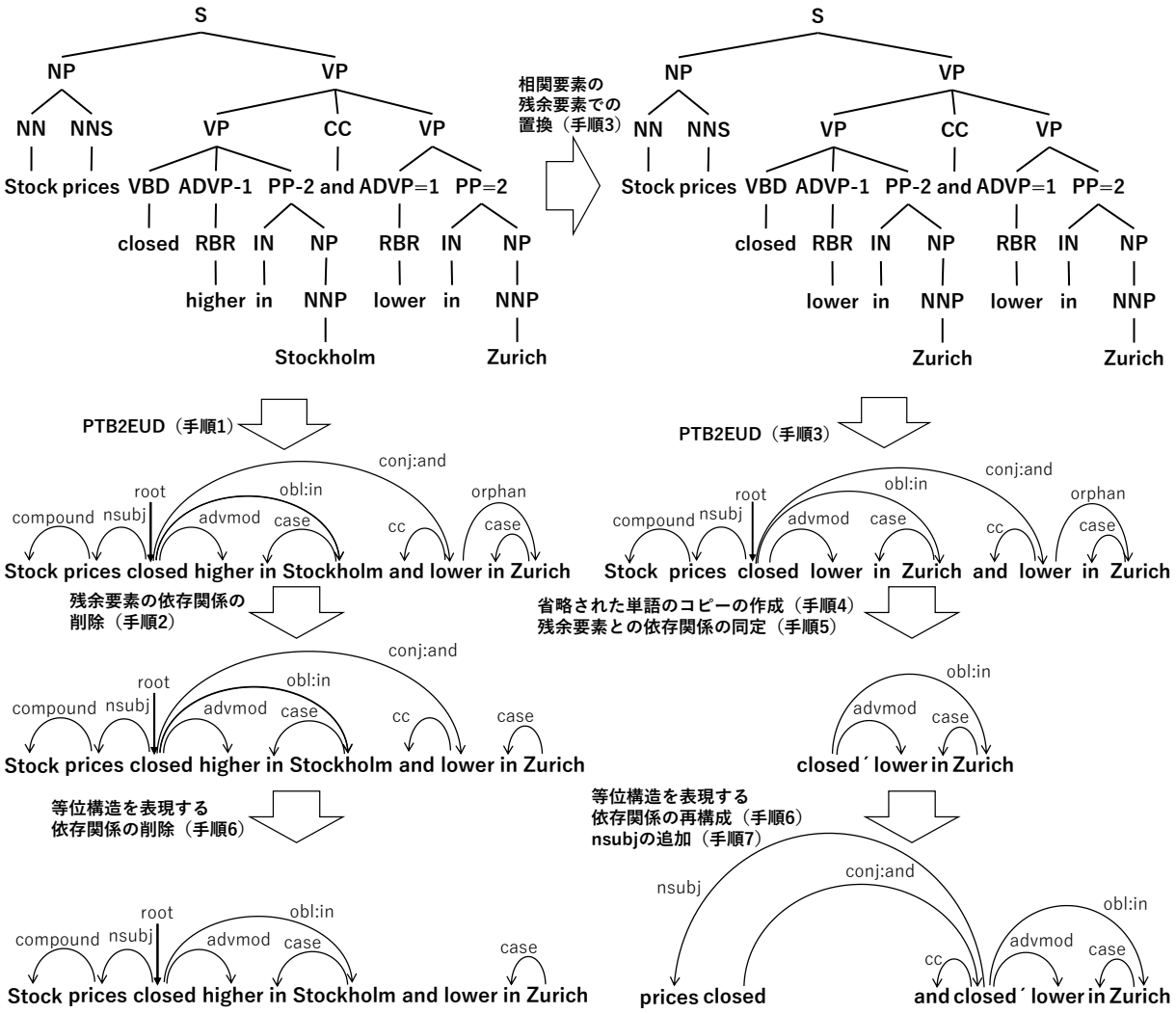

図 3　PTB から EUD への変換例

チングに基づき実行する．この変換では，空所化の情報については考慮されておらず，空所化構文について正しい EUD の依存構造を得ることはできない．

3　提案手法

本節では，空所化情報を考慮した PTB から EUD への変換手法を提案する．本手法では，各相関要素をそれに対応する残余要素で置き換えることにより，空所化構文の等位項における省略された単語を同定し，依存構造を構成する．

提案手法では，句構造 P を以下の手順により EUD の依存構造 E へと変換する．以下では，従来手法 [11] による PTB の句構造から EUD の依存構造への変換を，PTB2EUD と表記する．

1. 句構造 P を PTB2EUD により依存構造 $E_{-gapping}$ に変換する．
2. $E_{-gapping}$ から残余要素に関連する依存関係を取り除く．
3. 各相関要素を対応する残余要素で置き換え，結果として得られた句構造を PTB2EUD により依存構造 E' に変換する
4. E' 及び残余要素の情報に基づき省略された単語を同定し，それらの単語のコピーを作成する．
5. E' の情報に基づき，コピーした単語と残余要素間の依存関係を同定する．その結果を $E_{gapping}$ とする．
6. $E_{-gapping}$ から空所化構文の等位構造を表現する依存関係を削除する．$E_{gapping}$ の依存構造から等位項の head を決定し直し，その情報に基づき，削除した等位構造を表現する依存関係を再構成する．（$E_{gapping}$ に依存関係を追加する）
7. コピー元にのみ nsubj が存在する場合，それをコピー元の nsubj とする．nsubj 以外の core

表 1 実験結果

	UP	UR	LP	LR	SentenceAcc
Schuster et al. (COMPOSITE)	67.39	47.55	61.74	43.56	31.65
Schuster et al. (ORPHAN)	78.92	44.78	68.11	38.65	34.18
Kato et al.	82.46	57.67	78.07	54.60	40.51

argument についても同様の処理を行う．

8. $E = E_{gapping} \cup E_{-gapping}$

図 3 に変換の例を示す．手順 1 では，句構造 P から EUD の依存構造を従来の手法により求める．これにより，残余要素に関連しない依存関係については，従来手法に従うことになる．従来手法は空所化の情報を考慮していないので，手順 2 で残余要素に関連する依存関係を取り除く．手順 3 で得られる依存構造 E' は元の文の依存構造ではないが，これを元に，省略された単語や残余要素に関連する依存関係が同定される．手順 4 は，空所化構文における省略された単語を，PTB の空所化の情報に従って求める処理である．手順 5 によって，相関要素を含む等位項の head が変わる場合があるので，手順 6 において，変更された head に基づき等位構造の依存関係を付け直す．手順 7 は項が共有されている場合のための処理である．

以上の手順によって，PTB から EUD へ変換する際，PTB の空所化の情報に従って変換することができる．

4 評価実験

提案手法の応用例として，空所化構文を解析できる PTB に基づく句構造解析器と，依存構造解析器の性能比較実験を行った．PTB に基づく句構造解析器の解析結果を提案手法によって依存構造へ変換し，解析精度を依存構造ベースの評価尺度で評価した．依存構造解析器についても同じ評価尺度で評価し，それらを比較した．本実験は Schuster ら [15] の空所化構文の解析実験を参考とした．データセットには，Schuster らと同様のものを用いた．これは，(1) UD English Web Treebank v2.1 の 16,622 文と，(2) PTB や GENIA コーパス，Wikipedia の Gapping のページから集めた空所化を含むデータ 322 文に人手でアノテーション[1)]を付与したデータ[2)]からなる．実験における句構造解析には Kato らの手法 [10] を，依存構造解析には Schuster らの手法 [15] を用いる．句構造解析器の学習には，Schuster らの論文に対応する EWT のデータを用いた．提案手法における PTB から EUD への変換には，Stanford CoreNLP4.4.0[3)]を使用した．解析性能の評価では，コピーした単語を head とする依存関係を評価する．テストデータを用いて，依存関係の head 及び dependent の位置を組としたときの適合率・再現率（UP，UR），それに加えて，その依存関係の種類が正しいかどうか（LP，LR）を評価する．

表 1 に実験結果を示す．Kato らの手法は全ての指標において Schuster らの手法を上回る結果となった．このことから，Kato らの手法の方が空所化構文に対する解析性能が高いことがわかる．

このように提案手法を用いて変換を行うことで，句構造解析結果と依存構造解析結果を比較し，性能比較を行うことが可能となることが確認できた．

5 おわりに

本論文では，句構造から依存構造への変換において，空所化の情報を考慮した変換手法を提案した．空所化の情報を用いることで，PTB から EUD への変換において，省略された要素の同定，及び省略された要素と残余要素の間の依存関係を生成した．また，提案手法を用いて，空所化構文を解析する句構造解析器と依存構造解析器の性能比較実験を行い，空所化情報を考慮した性能比較が可能であることを確認した．

1） UD の依存関係に加えて，空所化構文の省略された要素と空所化構文に関連する依存関係がアノテーションされている

2） https://github.com/sebschu/naacl-gapping

3） https://stanfordnlp.github.io/CoreNLP/

謝辞

本研究は，一部，科学研究費補助金基盤研究（C）（No. 22K12148）により実施したものである．

参考文献

[1] Mark Steedman. **The Syntactic Process**. The MIT press, 2001.

[2] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In **Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)**, 2006.

[3] Jinho Choi and Martha Palmer. Guidelines for the clear style constituent to dependency conversion. Technical report, Technical report 01-12: Institute of Cognitive Science, University of Colorado Boulder, 2012.

[4] Xiaotian Zhang, Hai Zhao, and Cong Hui. A machine learning approach to convert CCGbank to Penn Treebank. In **Proceedings of the 24th International Conference on Computational Linguistics: Demonstration Papers (COLING 2012)**, pp. 535–542, 2012.

[5] Jonathan K. Kummerfeld, Dan Klein, and James R. Curran. Robust conversion of CCG derivations to phrase structure trees. In **Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (ACL 2012)**, pp. 105–109, 2012.

[6] Lingpeng Kong, Alexander M. Rush, and Noah A. Smith. Transforming dependencies into phrase structures. In **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)**, pp. 788–798, 2015.

[7] Young-Suk Lee and Zhiguo Wang. Language independent dependency to constituent tree conversion. In **Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016)**, pp. 421–428, 2016.

[8] 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. 313–330, 1993.

[9] John Robert Ross. **GAPPING AND THE ORDER OF CONSTITUENTS**, pp. 249–259. De Gruyter Mouton, 1970.

[10] Yoshihide Kato and Shigeki Matsubara. Parsing gapping constructions based on grammatical and semantic roles. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)**, pp. 2747–2752. Association for Computational Linguistics, 2020.

[11] Sebastian Schuster and Christopher D. Manning. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In **Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)**, pp. 2371–2378, 2016.

[12] Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. Genia corpus-a semantically annotated corpus for bio-textmining. **Bioinformatics**, Vol. 19 Suppl 1, pp. i180–182, 2003.

[13] Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. English web treebank. Technical report, **LDC2012T13, Linguistic Data Consortium**, 2012.

[14] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. Universal Dependencies v2: An evergrowing multilingual treebank collection. In **Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)**, pp. 4034–4043, 2020.

[15] 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 (NAACL-HLT 2018)**, pp. 1156–1168, 2018.