Vol.2013-NL-214 No.7
2013/11/14

# 依存構造解析における従属接続詞認識の効果

大内啓樹[†1]　増田優[†1]　金丸智史[†1]　松本裕治[†1]

従属節や埋め込み節を含む複雑な文を正確に解析することは困難である．本研究では，Penn Treebank コーパスの依存構造解析における従属接続詞表現の認識が持つ効果を調査した．我々は 2 つの文構造を結びつける表現を従属接続詞表現と広く定義し，S 構造の c-統御位置に現れ，SBAR 直下にある表現を従属接続詞表現として抽出した．さらに，テストデータに現れる従属接続詞表現を識別する分類器を作成し，英語依存構造解析においてそれらの自動でアノテーションされた属接続詞表現がどのように精度向上に寄与するかを調査した．

# Effect of Conjunctive Expression Recognition for Dependency Parsing

HIROKI OUCHI[†1]　YU MASUDA[†1]
SATOSHI KANAMARU[†1]　YUJI MATSUMOTO[†1]

Complex sentences that include subordinate or embedded clauses are difficult to parse correctly. This paper investigates the effect of subordinate conjunction recognition on the accuracy of word dependency parsing of the Penn Treebank corpus. We set a wide definition of subordinate conjunctive expressions in that we allow any expressions that connect two sentential structures. We extracted such expressions as those that appear at the c-commanding position of an S structure and are immediately dominated by an SBAR. We then construct classifiers to recognize them in test sentences and investigate how auto-tagged subordinate conjunctive expressions help to increase the accuracy of English dependency parsing.

## 1. はじめに

最近 10 年で, 依存構造解析精度向上のための研究が注力されて行われてきた．その結果，遷移型 [13), 8)]や，グラフ型 [7)]，高次グラフ型 [1), 2)]，ILP(整数線形計画法)型 [5), 10)]など多くの異なるタイプの依存構造解析アルゴリズムが提案されてきた．一方で，依存構造解析精度向上のためのさらなる問題がいくつか残っている．

問題の一つに，システムで使われる素性の粒度がある．多くの依存構造解析システムにおいてよく使われる素性は，細かすぎたり(例えば，語彙トークン)，粗すぎたり(例えば，品詞タグ)すると言える．このため，大規模コーパスから学習したクラスター素性が精度を向上させるのに有効である [2)]．CFG 型句構造解析において，シンボル細分化と呼ばれる技術が有効であることが知られている [6), 11)]．シンボル細分化とは，細かすぎたり粗すぎたりする非終端記号をそれらが現れる文脈情報を利用することによって，細分化し下位範疇化する技術である．依存構造解析では，クラスター素性の他に, 品詞タグの細分化が必要であると考えられる．

もう一つの問題は，同じ学習と推論のモデルが全ての統語的依存関係において使われていることである．単文構造内の局所的依存関係や，文間に跨りそれらを繋げる長距離依存関係は異なる性質を持っていると考えられる．言い換えれば，長文の依存構造解析におけるさらなる精度向上を達成するためにも，複文構造をより正確に解析することが求められる．文構造を繋げる典型的な表現が従属接続詞である．しかし，Penn Treebank(PTB)の品詞タグでは従属接続詞といくつかの品詞が区別されずに同じタグが付与されている．例えば，従属接続詞と前置詞は区別されておらず，どちらも同じ”IN”タグが付与されている．

本研究では，従属接続詞として機能する(一語，または複数語から形成される)従属接続詞表現の候補を PTB から抽出し，それらが文中に現れた際に従属接続詞として機能しているか否かを識別する．さらに，従属接続詞の同定が依存構造解析精度向上にどのように効果的かを調査する．この考えはどの依存構造解析アルゴリズムにも応用可能であるが，本研究では MST Parser (一次，二次モデル)を提案手法の評価のために使用する．

## 2. 関連研究

従属接続詞表現は，日本語や韓国語などの主要部後続言語の依存構造解析で特に調査されてきた．節間の依存関係の曖昧性解消のために，日本語従属節のスコープ選好性学習の統計モデルが提案されている [12)]．また，韓国語従属節依存構造の曖昧性解消のために木カーネルを使った類似したモデルも提案されている [4)]．日本語も韓国語も主要部後続言語であり，述語が従属節を形成する場合，接続表現が続く節(または文)の最後の要素が動詞などの述語となるので，節の最後の位置の同定は容易であり，節の終わりの表現間の依存関係同定に問題があるだけである．

一方, 英語における従属接続詞は節の先頭に現れるため，

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

従属節の始まりを見つけることは比較的容易である．しかし，従属節や主節の述語の位置の同定は容易なタスクではなく，節のスコープの同定も困難である．典型的な英語従属接続詞は，”although”，”when”，”if”などのように一語である．しかし，”even if”や”as though”などのように多くの複数語従属接続詞表現がある．そのような表現を認識することは依存構造解析精度に効果があることが知られている．例えば，複数語表現(MWE)を認識することによってスウェーデン語依存構造解析精度を向上させたという報告がある[9]．その報告では，従属接続詞に分類される MWE のみではなく，副詞，前置詞，限定詞，代名詞などの MWE が手動でアノテーションされたコーパスが使用されており，MWE の自動アノテーションを行った実験はなされていない．

英語には MWE アノテーションコーパスがないので，我々は最初に Penn Treebank における従属接続詞表現をどのように抽出するかを述べ，それらの表現の識別が依存構造解析精度向上にどのように寄与するかを述べる．それから，従属接続詞識別について説明し，最後に従属接続詞の自動アノテーションを伴った依存構造解析実験について示す．

## 3. 従属接続詞認識依存構造解析器

本節では，本研究における従属接続詞表現(SC)の抽出法と，SC が依存構造解析精度にどのような効果があるかを述べる．Penn Treebank(PTB)において，従属節や従属埋め込み文は図 1 が表しているような構造部分に現れる．

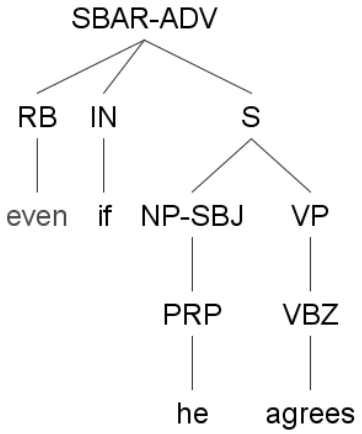


図 1　PTB における従属節の構文木

まず，PTB の全チャプターにおける SBAR-*と S の間(正確には，SBAR 構造直下かつ，S 構造の c-統御位置)に現れる全ての表現を抽出する．その際，空所(例えば，-NONE-)や記号は除外した．いくつかの表現は同じ品詞に属する語を交換可能な語として有する．本研究では，wh 疑問詞(what, who, when, how など)を含む SC における交換可能な語を，一つの品詞に抽象化し，一つの表現として結合して扱う．例えば，”in which”や”on which”などは”IN which”に，”how sweet”や”how much”は”how JJ”や”how RB”に，”whose portfolio”は”whose NN”というようにする．

表 1　SC 表現の出現頻度

| SC 表現候補 | SC 使用例 | 非 SC 使用例 |
|---|---|---|
| that | 8276 | 2046 |
| which | 2279 | 478 |
| who | 1901 | 1932 |
| when | 1444 | 1512 |
| … | … | … |
| IN which | 348 | 68 |
| whether | 328 | 7 |
| although | 324 | 5 |
| … | … | … |

抽象化の後，SC 候補として 441 個の表現を得た．しかし，これらの表現のすべてが PTB において SC として機能しているわけではない．表 1 は SC として使用されている表現とそうでないものの出現頻度を表している．4 節でそれらをどのように分類するかを説明する．

次に，ベースラインの依存構造解析と SC 認識依存構造解析の比較を行う．訓練データとして PTB のチャプター02-11 と 02-21 を，評価データとしてチャプター23 を使用した．SC 認識依存構造解析において，訓練データと評価データに現れる SC 表現の元々持つ品詞に”SC”という記号を結合し，従属節を表す新たな品詞として付与した．例えば，従属節である”if”の品詞は”IN”であるので，それを”SC”と結合し，”SC_IN”という新たな品詞として付与する．依存構造解析器には(１次・2 次)MST Parser を使った．結果を表 2 に示す．ただし，ラベルなし係り受け正解率(Unlabeled accuracy)とは各トークンの係り先が正しく同定できたものの割合であり，ラベルあり係り受け正解率(Labeled accuracy)とは各トークンの係り先と係り関係ラベルの両方が正解のものの割合を示す．

## 4. 従属接続詞の分類

本節では従属接続詞候補の分類実験に関して述べる．3 節でも触れたが，文中に現れた全ての SC 候補を SC として見なすことはできない．それらが SC ではない用法で現れることがあるからだ．出現した SC 候補が SC として使用されているか否かを自動分類するために，Support Vector Machine (SVM)で分類器を作成した．SC 候補のそれぞれの用例に対して，表 3 に示す素性を用いた．

表 2　ベースラインと SC 認識解析器の比較

| Setting | Parser | Training data | Unlabeled accuracy | Labeled accuracy |
|---|---|---|---|---|
| Baseline | MST-1 | 02-11 | 89.90 | 88.91 |
| | | 02-21 | 90.88 | 89.98 |
| | MST-2 | 02-11 | 90.88 | 89.84 |
| | | 02-21 | 91.91 | 90.98 |
| SC-aware | MST-1 | 02-11 | 90.30 | 89.34 |
| | | 02-21 | 91.37 | 90.51 |
| | MST-2 | 02-11 | 91.19 | 90.21 |
| | | 02-21 | 92.20 | 91.33 |

作成した分類器の性能は，PTB の 02-21 について SC 候補の自動分類を行った際には精度 93.72%, 再現率 90.94%, F 値 92.31 となった．これらの値はゴールドアノテーションと自動分類の SC 付与の結果を比較することで求めている. なお, 分類器は PTB のチャプターごとに作成してあり，それぞれの学習データにはチャプター02-21 から分類対象となるチャプターを除外して SC 表現とその素性を抽出したものを使用している．

SC 候補には SC としての用法のみをとるものがみられる．PTB においては次の語群 1 に示す 21 の表現が常に SC として使われていた．また語群 2 に示すのは PTB での出現のうち 95%が SC 用法であった表現である(括弧内の数は「SC 用法」/「非 SC 用法」の頻度を表す)．本研究では，これら 29 表現は分類器の対象とせず必ず SC として扱うこととした．

語群 1
even though, even when, to which, whenever, only if, as though, just because, just when, only when, whoever, especially if, some IN which, simply because, just how JJ, whereby, even while, three of whom, especially when, especially as, both IN which, whereas

語群 2
who(1904/28), when(1440/72), where(440/21), although(319/10), whether(325/10), unless(100/1), even if(86/2), as if(23/1)

また，SC として使われている回数より，そうでない回数が極端に多い表現を除外する目的で，次の条件を満たす表現のみを SC 表現候補として残した．

表 3　学習に使用した素性

| | |
|---|---|
| SC 表現自体 | 表層形，品詞，品詞細分類 |
| 文脈 | 前後 3 単語の表層形･品詞･品詞細分類，前後の bigram，前後の trigram，直後の動詞・助動詞との距離 |

(1) 正例が 10 以上ある．
(2) 正例は 10 未満だが，その割合が全体の 10%以上である．

例えば，”however”は SBAR 直下の c-統御位置に合計で 509 回出現しているが, SC としてはそのうちの 2 回しか現れず，上記の条件を満たさないので除外する．最終的に SC 候補として 389 個の表現を得た．

## 5. 従属接続詞自動分類を伴う依存構造解析

本節では，依存構造解析における SC 表現の自動分類の効果について述べる．PTB を係り受け関係に変換した CoNLL 形式のデータを使用し，実験を行った．訓練データとしてチャプター02-11 と 02-21 を, 評価データとしてチャプター23 を使用した．各データからベースラインのための SC 未付与のもの(Baseline)，SC のゴールドアノテーションであるもの(SC-aware)，SC を 4 節で作成した分類器を用いて自動付与したもの(SC-auto)の 3 種のデータセットを用意した．自動付与について詳細に述べると，4 節で述べた 29 の表現に SC 候補となる表現が含まれていれば，決定的に SC と見なし，それらに含まれていなければ，学習した分類器を適用し，SC か否かを判定した．

表 4 はそれぞれのシステム間の解析精度の比較結果を表している．また，ベースラインと各システムをマクネマー検定によって比較し, p 値が 5%未満のものを太字で示した．結果，MST パーザー1 次モデルでは，いずれの学習データを用いた場合でも，SC-aware と SC-auto は Baseline よりもラベルなし・あり係り受け正解率が高くなっており，有意水準 5%で有意差が認められる．また学習データの量の多い方が総じて高い性能が得られている．SC-auto の正解率が Baseline と SC-aware の正解率の間にあることは，SC の自動分類が必ず成功するわけではないことを反映していると考えられる．MST パーザー2 次モデルでの実験結果も 1 次モデルの際と同じ傾向を示しているが, SC 付与による正解率向上の幅が 1 次の場合より小さくなっている．これは変更したパーザーのアルゴリズム自体が長距離の依存関係をより適切に処理可能であるためだと考えられる．また, 2 次モデルでの SC-auto は，ベースラインより僅かに正解率が高いものの統計的有意差は認められなかった．

表 4　システム間の比較

| Parser | Traning data | Setting | Unlabeled accuracy | Labeled accuracy |
|---|---|---|---|---|
| MST-1 | 02-11 | Baseline | 89.90 | 88.91 |
| | | SC-aware | **90.30** | **89.34** |
| | | SC-auto | **90.04** | **89.01** |
| | 02-21 | Baseline | 90.88 | 89.98 |
| | | SC-aware | **91.37** | **90.51** |
| | | SC-auto | **90.99** | **89.94** |
| MST-2 | 02-11 | Baseline | 90.88 | 89.84 |
| | | SC-aware | **91.19** | **90.21** |
| | | SC-auto | 91.03 | 90.13 |
| | 02-21 | Baseline | 91.91 | 90.98 |
| | | SC-aware | **92.20** | **91.33** |
| | | SC-auto | 91.95 | 91.03 |

表 5 は係り先距離ごとの係り受け F 値を表している．表 5 の結果から，係り先距離が 3 以上のような場合により効果的だということが確認できる．つまり，1 節で述べた長距離依存問題に効果があることがわかる．また，ルート正解率に関しても，いずれのシステムも Baseline より高くなっていることが示されている．複文の構造がより正確に同定できていれば，ルートに係る語の候補の絞り込みに役立つと考えられる．

## 6. おわりに

本研究では，文中の従属接続詞の自動分類手法と，その分類が英語依存構造解析における解析精度にどのような影響を与えるかを調査した．我々の当初の目的は，節内構造と節間構造を識別し，それぞれに適した学習と推論のモデルを適用することであった．現段階では，節スコープを同定できていないため，本研究の拡張として，我々の従属接続詞分類手法と節スコープの同定を組み合わせることが考えられる．従属接続詞同定と節スコープ同定は相関するタスクなので，それらの問題を結合分析する必要がある．

もう一つの短期的な問題は，従属接続詞表現の完全なリストを作成することである．PTB はこのタスクのための言語資源として非常に有用であるが，他の言語資源からも従属接続詞表現を集める手法を探索する必要がある．そのような調査は，英語に限らず他言語にも応用可能である．

**謝辞**　本研究は，(独)情報通信研究機構の委託研究「知識・言語グリッドに基づくアジア医療交流支援システムの研究開発」の一環として実施した．

表 5　係り先距離ごとの比較: F 値

| Parser | Setting | Distance | | | | |
|---|---|---|---|---|---|---|
| | | root | 1 | 2 | 3-6 | 7 |
| MST-1 | Baseline | 95.86 | 96.32 | 94.28 | 89.99 | 88.56 |
| | SC-aware | 96.44 | 96.40 | 94.42 | 90.35 | 89.20 |
| | SC-auto | 95.99 | 96.35 | 94.29 | 90.10 | 88.60 |
| MST-2 | Baseline | 94.62 | 96.00 | 93.69 | 88.99 | 86.44 |
| | SC-aware | 95.45 | 96.20 | 94.05 | 89.41 | 87.28 |
| | SC-auto | 94.83 | 96.09 | 93.78 | 89.18 | 86.70 |

## 参考文献

1) Xavier Carreras.: Experiments with a Higher-Order Projective Dependency Parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, 957‑961 (2007).
2) Terry Koo, Xavier Carreras and Michael Collins.: Simple Semi-supervised Dependency Parsing. In *Proceedings of the 46th Annual Meeting of the ACL*, 595–603 (2008).
3) Terry Koo and Michael Collins.: Efficient Thirdorder Dependency Parsers. In *Proceedings of the 48th ACL*, 1-11 (2010).
4) Sang-Soo Kim, Seong-Bae Park and Sang-Jo Lee.: Dependency Analysis of Clauses Using Parse Tree Kernels. In *Proceedings of the 8th International Conference on Intelligent Text Processing and Computational Linguistics*, 218-228 (2007).
5) Andre Martins, Noah Smith and Eric Xing.: Concise Integer Linear Programming Formulations for Dependency Parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*, 342–350 (2009).
6) Takuya Matsuzaki, Yusuke Miyao and Jun'ichi Tsujii.: Probabilistic CFG with Latent Annotations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, 75–82 (2005).
7) Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic.: Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of HLT-EMNLP*, 523–530 (2006).
8) Joakim Nivre.: An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Workshop on Parsing Technologies (IWPT)*, 149–160 (2003).
9) Joakim Nivre and Jens Nilsson.: Multiword Units in Syntactic Parsing. *Workshop on Methodologies and Evaluation of Multiword Units in Real-world Applications, Workshop at LREC 2004,* 39–46 (2004).
10) Sebastian Riedel and James Clarke.: Incremental Integer Linear Programming for Non-projective Dependency Parsing. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 129–137 (2006).
11) Hiroyuki Shindo, Yusuke Miyao, Akinori Fujino and Masaaki Nagata.: Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 440–448 (2012).
12) Takehito Utsuro, Shigeyuki Nishiokayama, Masakazu Fujio and Yuji Matsumoto.: Analyzing Dependencies of Japanese Subordinate Clauses based on Statistics of Scope Embedding Preference. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 110–117 (2000).
13) Hiroyasu Yamada and Yuji Matsumoto.: Statistical Dependency Analysis with Support Vector Machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, 195–206 (2003).