実用的な構文解析器の開発

辻井潤一

情報理工学系研究科コンピュータ科学専攻

概 要

言語学に基づく構文解析を，実用的な言語処理アプリケーションに適用することを目指し，そのために必要な要素技術の開発を行なっている．本稿では特に文法の自動的な体系化と，構文解析の高速化について報告する．

1 はじめに

本研究では，言語学に基づく深い構文解析を，実用的な自然言語処理アプリケーションへ適用することを目指す．そのためには，最先端の言語理論に基づく文法の効率的な開発，またそれを利用した高速な構文解析アルゴリズムなどの要素技術が必要不可欠である．本稿では，文法開発における自動的な文法体系化，および統計モデルによる枝刈りを応用した構文解析の高速化手法の研究について報告する．

近年，大規模な文法を効率的に開発する手法として，注釈付きコーパスから自動的に文法を獲得する手法が提案されている．コーパスに現出した人間の言語直感を利用して，そこから文法を自動獲得することにより，言語学に基づき，かつ頑健な文法を効率的に開発することが可能となった．しかしながら，自動獲得した文法は系統的な体系化がされていないため，人間には把握しにくく，そのため人間の手により文法を拡張・改良していくことが困難となっている．また，文法が学習コーパスに強く依存したものになるという問題もある．本研究では，このような問題を解決するために，文法を自動的に体系化する手法を提案する．特に，本稿では構造的クラスを自動的に獲得する手法について報告する [1] (2 節)．

上記のアプローチにより言語学に基づく大規模な文法の開発が可能になり，非局所的な深い言語学的関係を扱える HPSG, LFG, CCG 等の文法枠組によって与えられる構文解析結果とそれらに割り当てられる評価値による曖昧性解消が提唱されている [2]．しかし，これらの評価値付けモデルでは部分解析結果の非局所性から評価値を構成的に計算できないため，部分解析結果に評価値が割り当てられていることを前提とする既存の枝刈り手法を単純には導入することができない．本研究ではこの問題を解決するため，非局所的な関係を含む部分解析結果の評価値を構成的に計算する手法を提案する [3]．本手法は部分解析結果中の構造が確定した部分構造の重みのみを計算に用いることで部分解析結果にも評価値を割り当て，評価値の構成的計算を可能とする (3 節)．

2 語彙化文法の自動的な体系化

言語学に基づく多くの文法枠組では，単語固有の文法的特徴を語彙項目に記述し，語彙項目を (i) 統語的クラス（形容詞，他動詞や二重他動詞）と (ii) 構造的クラス（受動文・命令文など）の組み合わせにより表現することで，大規模な文法を体系的・効率的に表現する試みが行われてきた．このうち，統語的クラスの自動分類は既に提案されている [4]．そこで本研究では，構造的クラスを自動的に獲得することを目指す．

本研究では，各語彙項目は平叙文の語彙項目から統語的変形により生成されていると考え，この統語的変形を語彙項目の構造の差異という形で得

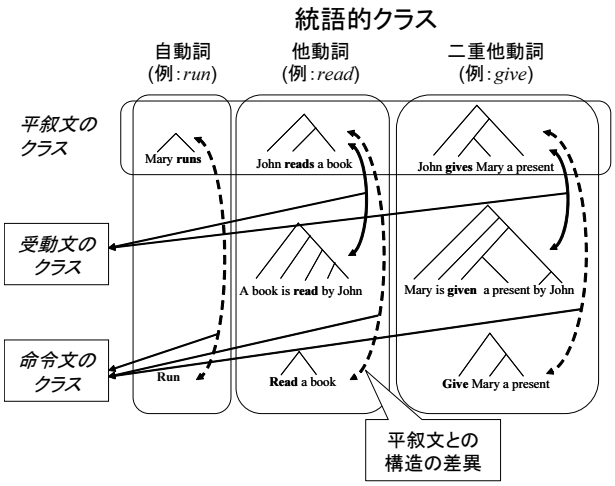

図 1: 構造的クラスの自動獲得

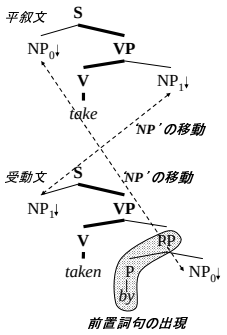

図 2: 受動文の語彙項目の構造的素性

て，その差異に基づき分類を行う（図 1）．図 2 は LTAG において受動文の語彙項目と平叙文の語彙項目との構造的差異を図示したものである．図中で斜線で示された構造は受動文の語彙項目に新たに現れた構造 (出現素性) であり，点線で示された対応は下位範疇化要素の順列の変化 (移動素性) を表している．我々は語彙項目と平叙文の語彙項目との間の構造的差異を，上記 2 種類の構造的素性でとらえ，その類似性によるクラスタリングを行なうことで，構造的クラスを自動獲得する．

本手法を語彙化文法の一つである LTAG を対象として実装した．入力として大規模 LTAG 文法である XTAG 英文法の tree family と呼ばれる動詞の統語的クラス 57 個とそれに属する 1,008 個の語彙項目を入力として用いた．その結果，63 種類

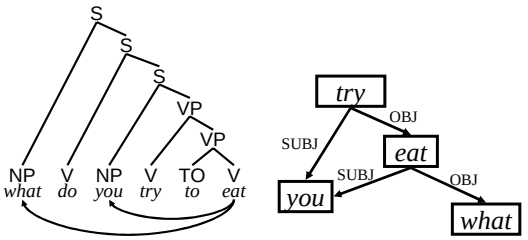

図 3: 文 “What do you try to eat?” に対する構文木と項構造

の出現素性，5 種類の移動素性を抽出し，134 個のクラスを得た．得られた 134 のクラスと XTAG 英文法の平叙文以外の 73 の構文的クラスについて，本手法で得られたクラスの中に一致するものがあるかどうかを調べた．その結果，人手による構文的クラスのうち 34 (46.6%) は得られたクラスの中に一致するものがあり，31 (42.5%) は得られたクラスの中に細分化されたクラスがあった．残りの 8 (10.9%) については得られたクラスの中では一部が他のクラスと混ざっていた．この結果から，我々の手法により人間の直感と一致した，もしくはより細分類された構造的クラスを自動的に獲得することが可能であることが示された．

3 非局所的な評価値の構成的な計算

本節では非局所的な関係を含む部分解析結果の評価値を構成的に計算する手法について報告する．今回は特に，HPSG に基づく文法である XHPSG 文法 [5] による構文解析によって得られた項構造上の確率モデル [2] に本手法を適用して枝刈り手法を導入する実験を行い，その妥当性と有効性を検証した．項構造とは語と語の依存関係を表現した意味表現のひとつである．図 3 は *“What do you try to eat?”* に対する構文木とそれに対応する項構造である．構文木では *you* と *eat*，*what* と *eat* の非局所的な関係は表現できないが，項構造では表現されている．項構造上の確率モデルは項構造 A の評価値 $\zeta_a(A)$ を以下のように与える: $\zeta_a(A) = \prod_{S \in \mathcal{S}(A)} w(S)$. $\mathcal{S}(A)$ は項構造 A の部分構造，すなわち部分グラフの集合であり，$w(S)$ は

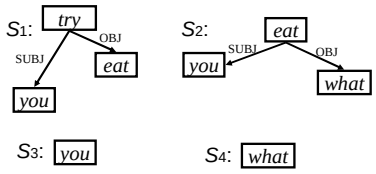

図 4: 図 3 の項構造の部分構造 S_1, S_2, S_3, S_4

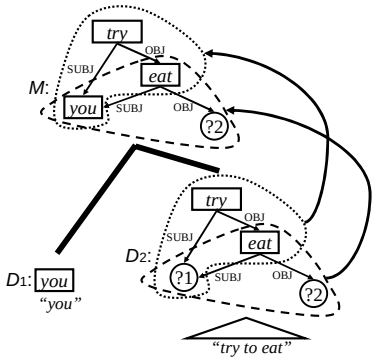

図 5: *"you try to eat"* の項構造の構築

確率モデルによって定義される部分構造 S の重みである．

図 5 は *"you"* と *"try to eat"* の項構造を組み合わせて *"you try to eat"* の項構造を構築する様子である．図 5 の *"try"* と *"eat"* を起点とする二つの点線に囲まれた部分構造は子 D_2 から親 M に伝播するが，文法適用の結果として構造が変化していることに注意されたい．特に *"eat"* を起点とする部分構造の変化は，非局所的な関係によりグラフの任意の部分が変化しうることを示唆するため問題となる．単純に全ての部分構造の重みの積を評価値として用いると D_2 と M の評価値計算でそれぞれ変化前後の部分構造の重みを乗算してしまうため，D_2 の評価値を M の評価値の計算に再利用できない．従って，このような評価値の構成的計算を行うためには部分構造が変化してしまう問題に対処しなければならない．

この問題を解決するため，本手法では構造が確定した部分構造の重みのみを用いて部分解析結果の評価値を計算する．一般的に，本手法による部分解析結果 R の評価値 $\zeta'(R)$ は以下の計算で求める．

$$\zeta'(R) = g(\mathcal{U}(\mathcal{S}(R))) \quad (1)$$
$$g(\mathbf{S}) = w(S_1) \circ w(S_2) \circ ... \circ w(S_m)$$
$$\text{where } \mathbf{S} = \{S_1, S_2, ..., S_m\}$$

ただし，$\mathcal{U}(\mathcal{S}(R))$ は部分構造の集合 $\mathcal{S}(R)$ 中の確定した部分構造のみからなる集合，$w(S)$ は部分構造 S の重み，$\circ$ は元々の評価値計算に使われている二項演算子を表す．項構造上の確率モデルの場合，項構造 A の (1) 式による本手法での評価値は $\zeta'_a(A) = \prod_{S \in \mathcal{U}(\mathcal{S}(A))} w(S)$ となる．親 M とその子に関して本手法における評価値 $\zeta'(M)$ の計算式 (1) は以下のように変形できる．

$$\zeta'(M) = g(\mathcal{NS}(M)) \circ \zeta'(D_1) \circ ... \circ \zeta'(D_n)$$

ただし，$\mathcal{NS}(M)$ は M で新規に構造が確定した部分構造の集合である．$\zeta(M)$ の代わりに $\zeta'(M)$ を用いることで評価値の構成的計算が可能となる．

本手法に基づいた評価値計算による枝刈りの効率を見るため，以下の 3 種類の構文解析器が構文解析中に生成する項構造数の比較を行った．

baseline CKY スタイルの構文解析によって全ての項構造を生成し，全く枝刈りを行わない構文解析器

best-first parser baseline に best-first parsing による枝刈りを実装した構文解析器

beam thresholding baseline に beam thresholding による枝刈りを実装した構文解析器

大規模な英文法である XHPSG 文法 [5] と項構造上の確率モデルの小規模なもの [2] を使用し，本手法の効果を観察した．テストデータとしては，Penn Treebank の ATIS コーパスの中で文長 20 語以下で XHPSG 文法により文と解析されたもの 251 文を使用した．

baseline と best-first parsing による結果は表 1 のようになった．この結果によると best-first parsing では殆ど枝刈りが出来ていないことが分かる．これは，多くの部分構造を持つ長い句に対する部

parser	項構造数
baseline	877911
best-first parser	861289

表 1: baseline と best-first parser における項構造数の比較

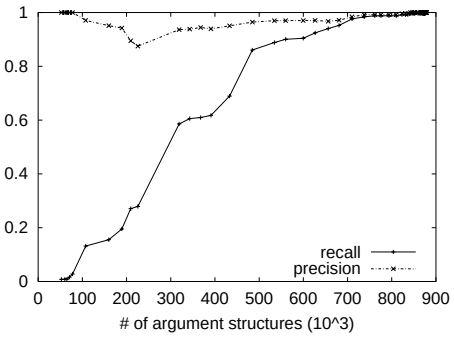

図 6: beam thresholding における項構造数（横軸）と precision, recall（縦軸）の関係

分解析結果の評価値が劇的に低くなってしまうために，best-first parsing が幅優先探索のように動いてしまったためと考えられる．また，baseline と best-first parsing では出力される項構造が完全に一致した．これは，本手法による評価値計算が妥当であることを示している．

一方，beam thresholding による枝刈りは最大の評価値を持つ項構造の出力を保証しないため，閾値を変化させて precision/recall と項構造数の関係を見ることで高精度を保持したときの枝刈りの効果を検証した．

$$precision = \frac{N_{agree}}{N_{beam}},\ recall = \frac{N_{agree}}{N_{base}}$$

ただし N_{agree} は baseline と beam thresholding の出力が一致した文数，N_{beam} は beam thresholding が出力した項構造の数，N_{base} は baseline が出力した項構造の数である．図 6 に示すとおり，precision は最低でも 87.5%であり，beam thresholding の出力は項構造上の確率モデルを用いた曖昧性解消の結果として信頼できる．また，recall が 90% 時の beam thresholding が生成する項構造数は baseline の 63.8%であり，ある程度の枝刈りができているといえる．以上の結果より，本手法によって非局所的な関係を含む構文解析結果を出力する構文解析器に既存の枝刈り手法を導入することが可能であることが示された．

4 おわりに

本稿では，文法を自動的に体系化するために構造的クラスの分類を行なう手法，及び，構文解析の高速化のために統計モデルによる枝刈りを行なう手法について報告した．前者は言語学に裏付けされた頑健な文法を開発するために貢献し，後者はそれを用いた構文解析を様々な実用的アプリケーションに応用することを可能とする．今後は，これらの手法を統合し，実用的な構文解析器を実現することを目指す．

参考文献

[1] 大内田賢太，吉永直樹，二宮崇，宮尾祐介，辻井 潤一. 語彙化文法における語彙項目の構造的特徴に基づく自動分類. In **言語処理学会第９回年次大会論文集**, 2003.

[2] 宮尾祐介，辻井潤一．確率付き項構造による曖昧性解消. In **言語処理学会第 *6* 回年次大会発表論文集**, pages 495–498, 2000.

[3] 坂尾要祐，宮尾祐介，辻井潤一．構文解析における非局所的な評価値の構成的計算. In **言語処理学会第９回年次大会論文集**, 2003.

[4] Tadayoshi Hara, Yusuke Miyao, and Jun'ichi Tsujii. Clustering for obtaining syntactic classes of words from automatically extracted ltag grammars. In *Proc. of TAG+6*, pages 227–233, 2002.

[5] Y. Tateisi, K. Torisawa, Y. Miyao, and J. Tsujii. Translating the XTAG English grammar to HPSG. In *Proc. of TAG+4 Workshop*, pages 172–175, 1998.