

構文解析における非局所的な評価値の構成的計算

坂尾 要祐† 宮尾 祐介† 辻井 潤一‡

† 東京大学大学院情報理工学系研究科コンピュータ科学専攻

‡ CREST, 科学技術振興事業団

{sakao,yusuke,tsujii}@is.s.u-tokyo.ac.jp

1 はじめに

近年、非局所的な深い言語学的関係を扱える HPSG [9] や LFG [5] 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 [11, 1]。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。従ってこれらの評価値付けモデルを用いた構文解析器を高速化するのは現時点では困難である。

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。非局所的な関係を含む部分解析結果に対する評価値が構成的に計算できないのは、非局所的な関係に対応する部分構造が構文解析中に変化することが原因となっている。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。本手法により部分解析結果に割り当てられる評価値を用いて、best-first parsing [7] や beam thresholdong [2, 3] 等の既存の枝刈り手法を導入することができる。

本手法はいくつかの条件を満たす任意の文法と評価値付けモデルに対して適用可能である。今回は特に、HPSG に基づく文法である XHPSG 文法 [10] 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 [11] 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。その結果、本手法による評価値計算の妥当性が示され、さらに beam thresholding によってある程度の枝刈りに成功した。本稿ではまだ項構造上の小規模な確率モデルを用いた実験のみに留まっているため、他の評価値付けモデルに本手法を適用する実験も行う必要があると考えられる。

第2節では構文解析結果と評価値付けモデルの例として項構造とその上に定義される確率モデルを導入し、それらを例として評価値の構成的計算を行う際に生じる問題について述べる。第3節では本稿の提案する評価値の構成的計算手法について述べる。第4節では、本手法による評価値計算を応用した枝刈り手法の有効性を、実験を通して検証する。

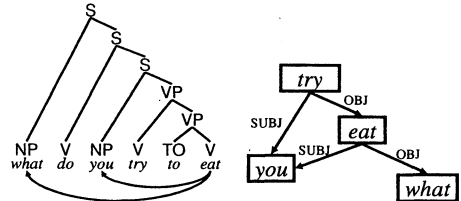


図1: 文 “What do you try to eat?” に対する構文木と項構造

2 構文解析結果と評価値付けモデル

本節では構文解析結果と評価値付けモデルの例として、項構造とその上に定義される確率モデル [11] について述べ、そこで評価値の構成的計算を行うと生じる問題について述べる。

項構造とは語と語の依存関係を表現した意味表現 [4] のひとつである。項構造は、語に対応するノードを関係に対応するラベル付き有効枝で結んだグラフ構造により表現できる。図1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 [11] はこの項構造の上に定義される評価値付けモデルである。この確率モデルにおいて項構造 A の評価値 $\zeta_a(A)$ は以下のように計算される。¹

$$\zeta_a(A) = \prod_{S \in S(A)} w(S)$$

$S(A)$ は項構造 A の部分構造、すなわち部分グラフの集合である。また、 $w(S)$ は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。ここでは、説明を簡単にするために各ノードとそれを起点とする有向枝からなる部分構造のみを評価値計算に用いると仮定する。この場合、図1の項構造の評価値は図2に示された部分構造の重みの積として計算される。図2の S_2 が “eat”, “you”, “what” の3項関係を示す部分構造であるため、項構造上の確率モデルはこの非局所的な関係も捉えることができている。

¹項構造の確率を計算するためにはさらに $\zeta_a(A)$ をモデルによって決まる定数 Z_h で割って0以上1以下に正規化する必要があるが、評価値としては正規化前の値を用いても問題はない。

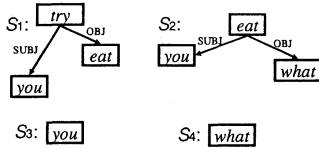


図 2: 図 1 の項構造の部分構造 S_1, S_2, S_3, S_4

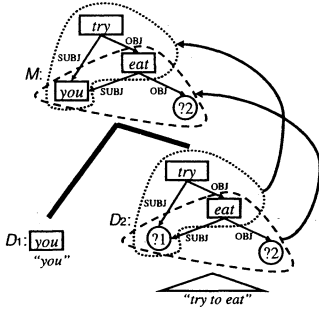


図 3: “you try to eat” の項構造の構築

次に、このような評価値を構成的に計算しようとする際に発生する問題について述べる。図 3 は “you” と “try to eat” の項構造を組み合わせ “you try to eat” の項構造を構築する様子である。図 3 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。

3 評価値の構成的計算

前節では、未確定な部分構造の存在により部分解析結果の評価値が構成的に計算できないという問題点を示した。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。

$$\begin{aligned} \zeta'(R) &= g(U(S(R))) \\ g(S) &= w(S_1) \circ w(S_2) \circ \dots \circ w(S_m) \\ \text{where } S &= \{S_1, S_2, \dots, S_m\} \end{aligned} \quad (1)$$

ただし、 $U(S(R))$ は部分構造の集合 $S(R)$ 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

```

INPUT: 部分解析結果  $M$  とその子  $D_1, D_2, \dots, D_n$ 
OUTPUT:  $M$  の評価値  $\zeta'(M)$ 
procedure fom_of_mother( $(M, [D_1, D_2, \dots, D_n])$ )
begin
   $S_M := C[D_1] \cup C[D_2] \cup \dots \cup C[D_n]$       (1)
   $S_u := U(S_M)$                                 (2)
   $c[M] := S_M - S_u$                              (3)
   $\zeta'(M) := g(S_u) \circ \zeta'(D_1) \circ \dots \circ \zeta'(D_n)$  (4)
  return  $\zeta'(M)$ 
end

```

図 4: 部分解析結果の評価値を構成的に計算するアルゴリズム

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。第 2 節で導入した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 $\zeta'_a(A) = \prod_{S \in U(S(A))} w(S)$ となる。

この手法を適用するためには、部分解析結果を構築する文法と評価値付けモデルがそれぞれ以下の条件を満たしている必要がある。

文法の条件 構文木中の任意の部分解析結果 M とその子の集合 $\{D_1, D_2, \dots, D_m\}$ について以下の命題が成り立つ。

$$S(M) = \bigcup_{i=1, \dots, m} S(D_i)$$

評価値付けモデルの条件 構文解析結果 R の評価値 $\zeta(R)$ が以下のような計算式で求まる。

$$\begin{aligned} \zeta(R) &= g(S(R)) \\ g(S) &= w(S_1) \circ w(S_2) \circ \dots \circ w(S_m) \\ \text{where } S &= \{S_1, S_2, \dots, S_m\} \end{aligned}$$

ただし $w(S)$ は部分構造 S の重み、 \circ は交換則と結合則を満たす 2 項演算子を表す。 w, \circ は評価値付けモデルによって決まる。

これらは直感的には「親は全ての子の部分構造を含む」、「解析結果の評価値はその部分構造の重みから計算される」ということを示している。この条件の下で、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。

$$\zeta'(M) = g(NS(M)) \circ \zeta'(D_1) \circ \dots \circ \zeta'(D_n)$$

ただし、 $NS(M)$ 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。ここで新規に確定した部分構造を効率的に探索する必要がある。本手法では、新規に確定した部分構造の探索範囲を制限するために、部分解析結果毎に未決定の部分構造の集合を格納する配列 $c[R]$ を用意する。親 M において新規に確定した部分構造の探索は、集合 $c[D_1] \cup c[D_2] \cup \dots \cup c[D_n]$ に制限される。

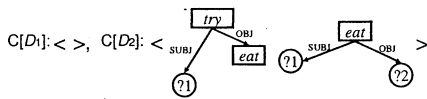


図 5: D_1, D_2 の未確定の部分構造の集合 $C[D_1], C[D_2]$

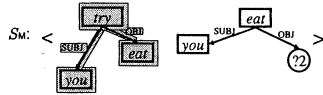


図 6: 図 4 の (1) 直後の部分構造の集合 S_M

図 4 の *fom_of_mother* は部分解析結果 M とその子 D_1, \dots, D_n を受け取り、 M の評価値 $\zeta'(M)$ を返す関数である。まず子の未確定の部分構造の集合 $C[D_1], \dots, C[D_n]$ の和集合を親 M でまだ評価値の計算に使われていない部分構造の集合 S_M とする (図 4 の (1))。次に、 S_M を親 M で新規に確定した M の部分構造の集合 S_u と未確定の部分構造の集合 $C[M]$ に分割する (図 4 の (2),(3))。最後に S_u 中の部分構造の重みと子 D_1, \dots, D_n の評価値から親 M の評価値を計算する (図 4 の (4))。

図 3 の親 M と子 D_1, D_2 を *fom_of_mother* の入力とした場合の動作例を示す。図 5 は娘 D_1, D_2 の未確定の部分構造の集合であり、これを連結した、 S_M は図 6 のようになる。図 6 の影のついた部分構造は、今回の項構造の構築によって新規に構造が確定した部分構造である。これを新規に確定した部分構造の集合 S_u と未確定の部分構造の集合 $C[M]$ に分割すると図 7 のようになる。最終的に S_u の要素の重みと $\zeta'_a(D_1), \zeta'_a(D_2)$ の積として親 M の評価値 $\zeta'_a(M)$ が計算される。

4 実験

本節では本手法による評価値計算を用いた枝刈りを実装して実験を行い、その動きを観察する。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、我々は以下の 3 種類の構文解析器を実装して構文解析中に生成される edge 数と項構造数の比較を行った。edge とは部分解析結果を統語的性質ごとにまとめたものであり、1 つの edge に同じ統語的性質を持つ複数の項構造が対応する。

baseline CKY [6] 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 [11]

best-first parser baseline に best-first parsing [7] による枝刈りを実装した構文解析器

beam thresholding baseline に beam thresholding [2, 3] による枝刈りを実装した構文解析器

文法は HPSG [9] に基づく大規模な英文法である XHPSG 文法 [10] を使用した。評価値は、項構造上の確率モデル

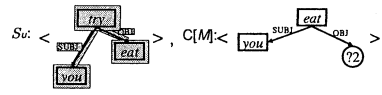


図 7: 図 4 の (2),(3) 直後の集合 $S_u, C[M]$

parser	edge 数	項構造数
baseline	111445	877911
best-first parser	110869	861289

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

で各部分構造の重みを 0 以上 1 以下に正規化したものを用いた。本手法は本来ならば 3 節で述べた条件を満たす任意の評価値付けモデルに対して適用可能だが、今回は項構造上の確率モデルの小規模なもの [11] を使用し、本手法の効果を観察した。この確率モデルは ATIS コーパス [8] の 303 文から取得した項構造から学習されたものである。テストデータとしては、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、部分構造の重みが 0 以上 1 以下に正規化されているので、多くの部分構造を持つ長い句に対する部分解析結果の評価値が劇的に低くなってしまいうために、best-first parsing が幅優先探索のように動いてしまっているためと考えられる。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障しないため、閾値を変化させて精度と edge 数、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。ここで精度の指標として用いる precision と recall とは以下のよう

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

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 が出力した意味構造の数、 N_{base} は baseline が出力した意味構造の数である。edge 数と精度の関係は図 8、項構造数と精度の関係は図 9 のようになった。図 8 より precision は最低でも 87.5% であり、beam thresholding が出力する項構造の多くは baseline と同じなので 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 edge 数は baseline の 97.2% であり、recall を高く維持しようとする edge はほとんど枝刈りできていない。一方、図 9 より recall が 90% 時の beam thresholding が生成する項構造数は baseline の

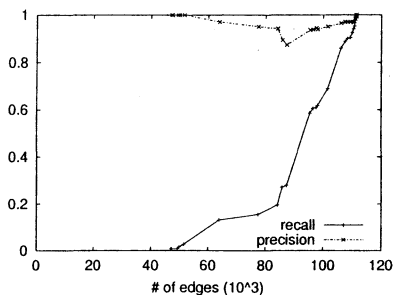


図 8: beam thresholding における edge 数 (横軸) と precision, recall (縦軸) の関係

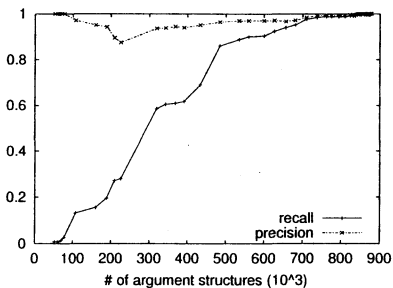


図 9: beam thresholding における項構造数 (横軸) と precision, recall (縦軸) の関係

63.8%であり、項構造に関してはある程度の枝刈りができている。

5 おわりに

本稿では、非局所的な関係を含む構文解析結果に割り当てられる評価値を構成的に計算する手法を提案した。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 best-first parser や beam thresholding 等の既存の枝刈り手法を導入できるようになった。さらに本手法に基づいた評価値計算を用いた枝刈り手法を実装し、その動きを観察した。その結果、best-first parser ではほとんど枝刈りができなかったが beam-thresholding では高精度を保ったままある程度の枝刈りに成功した。

本手法は項構造上の確率モデルに限らず、第 3 節で導入した条件を満たす任意の文法と評価値付けモデルに適用可能である。本手法の性質をより明らかにし、本手法を用いた枝刈りをより効率的にしていくには、様々な評価値付けモデルに対する本手法の動きを観察すること、本手法を用いた枝刈りを効率的に行える評価値付けの枠組みの構築が必要であると考えている。

参考文献

- [1] S. Clark, J. Hockenmaier, and M. Steedman. Building deep dependency structures with a wide-coverage CCG parser. In *Proc. of the 40th ACL*, pages 327–334, 2002.
- [2] M. J. Collins. A new statistical parser based on bigram lexical dependencies. In *Proc. of the 34th ACL*, pages 184–191, 1996.
- [3] J. Goodman. Global thresholding and multiple-pass parsing. In *Proc. of the Second Conference on Empirical Methods in Natural Language Processing*, pages 11–25, 1997.
- [4] R. Jackendoff. *Semantic Structures*. The MIT Press, 1990.
- [5] R. M. Kaplan and J. Bresnan. Lexical-functional grammar: A formal system for grammatical representation. In *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, 1982.
- [6] T. Kasami. An efficient recognition and syntax algorithm for context-free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Lab., 1965.
- [7] M. Kay. Algorithm schemata and data structures in syntactic processing. In B. J. Grosz, K. Sparck Jones, and B. L. Webber, editors, *Natural Language Processing*, pages 35–70, 1986.
- [8] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.
- [9] C. Pollard and I. A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, 1994.
- [10] 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.
- [11] 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.