複数語ごとの入力に対する漸進的構文解析

森 大輔† 加藤 芳秀† 松原 茂樹‡ 稲垣 康善†

†名古屋大学大学院工学研究科 ‡名古屋大学言語文化部

1 はじめに

漸進的構文解析は, 自然言語文の入力途中の段階で随時, それまでの入力に対する構文構造を作成する枠組であり, 音声を聴きながら理解するシステムの要素技術の一つである. これまでにいくつかの漸進的構文解析手法が提案されているものの [3, 4, 1], これらの手法はいずれも語単位で解析を実行するため, 処理の効率はよくない.

漸進的構文解析を応用するという観点からみた場合, 解析が語単位である必要は必ずしもない. 複数語ごとに解析を実行すれば作成すべき構文構造が減少するため, 処理の向上が見込まれる.

そこで本稿では, 複数語ごとの入力に対する漸進的構文解析手法を提案する. 本手法では, 漸進的構文解析の一手法である漸進的チャート解析 [3] に, 不要な構造を解析途中で取り除く操作を加えることにより, 解析の効率化を実現する. また, 解析実験による本手法の評価について報告する.

2 漸進的構文解析とその効率

2.1 漸進的チャート解析

漸進的チャート解析 [3] は, 通常の上昇型チャート解析に, **1) 活性弧に文法規則を適用する操作**, 及び, **2) 活性弧同士を結合する操作**, を導入している. これらの操作により, 語が入力されるたびにその時点までの入力に対する構文構造の作成が可能となる. 例えば, 英語文

(1)"Ken's brother saw her in the park"

を解析する場合, まず, "Ken's brother" が入力された時点で, 図 1の文法と辞書を用い,

(2) $[[[\text{Ken's}]_{\text{adj}}[\text{brother}]_{\text{n}}]_{\text{np}}[?]_{\text{vp}}]_{\text{S}}$

を作成する. 次の語 "saw" が入力された段階で, "saw" に対する構造 "$[[\text{saw}]_{\text{vt}}[?]_{\text{np}}[?]_{\text{pp}}]_{\text{vp}}$" を作成し, これを (2) に結合することにより, "Ken's brother saw" に対する構造

(3) $[[[\text{Ken's}]_{\text{adj}}[\text{brother}]_{\text{n}}]_{\text{np}}[[\text{saw}]_{\text{vt}}[?]_{\text{np}}[?]_{\text{pp}}]_{\text{vp}}]_{\text{s}}$

を作成する. "saw" の主語が "Ken's brother" である, などといった構成素間の関係を早い段階で明らかにできるという特長がある.

2.2 漸進的構文解析の効率

漸進的チャート解析は, 語単位でそれまでの入力に対する構文構造を作成するため, 解析の効率はよくない [2]. 例えば, 英語文 (1) の "Ken's brother" の解析が終了した時点で, 新たに "saw her in" を解析する場合, "Ken's brother saw her in" に対する構造

文法	辞書
s → np vp	pron → Ken / her
np → pron	det → her /Ken's
np → det n	n → brother / park
pp → p np	p → in
vp → vt np pp	vt → saw

図 1: 文法と辞書

(4) $[[[\text{Ken's}]_{\text{adj}}[\text{brother}]_{\text{n}}]_{\text{np}}[[\text{saw}]_{\text{vt}}[\text{her}]_{\text{pron}}]_{\text{np}}[[\text{in}]_{\text{p}}[?]_{\text{np}}]_{\text{pp}}]_{\text{vp}}]_{\text{s}}$

を作成しさえすればよい. ところが, 漸進的チャート解析では, "saw" の入力に対する構文構造として (3) を, また "her" に対して

(5) $[[[\text{Ken's}]_{\text{adj}}[\text{brother}]_{\text{n}}]_{\text{np}}[[\text{saw}]_{\text{vt}}[[\text{her}]_{\text{det}}[?]_{\text{n}}]_{\text{np}}[?]_{\text{pp}}]_{\text{vp}}]_{\text{s}}$

(6) $[[[\text{Ken's}]_{\text{adj}}[\text{brother}]_{\text{n}}]_{\text{np}}[[\text{saw}]_{\text{vt}}[[\text{her}]_{\text{pron}}]_{\text{np}}[?]_{\text{pp}}]_{\text{vp}}]_{\text{s}}$

を作成する. このうち, (6) は次に前置詞句 (pp) が入力されることを示しており, (4) の作成に必要な構造である. 一方, (5) は次に名詞 (n) が入力されることを示しているものの, 次の語が "in" であることから, それは不要な構造であるといえる. このように, 入力された複数語全体の情報を活用することにより不要な弧の作成を避けることが可能となる.

3 複数語全体を活用する漸進的チャート解析

3.1 不要な活性弧の除去

本節では, 複数の語が入力された段階で, 構文構造を効率よく作成する手法を提案する. 本手法では, 従来の漸進的チャート解析の操作 1), 2) に先だって, 複数語全体の情報を活用することにより, 不要な活性弧を取り除く. そのアイデアは以下の通りである. ある時点までの入力に対する構文構造は, 必ずその時点までの入力の最後の単語にまたがる弧の構文構造を構成要素として含むことになる. 例えば, "Ken's brother saw her in" に対する構文構造 (4) は, "in" にまたがる弧の構文構造 "$[[\text{in}]_{\text{p}}[?]_{\text{np}}]_{\text{pp}}$" を含む. このことから, その時点の最後の単語にまたがる弧と結合できない弧は, その後の解析に必要ないといえる. すなわち, 構造 "$[[\text{her}]_{\text{adj}}[?]_{\text{n}}]_{\text{np}}$" をもつ弧は, "in" にまたがる弧と結合できないため, "Ken's brother saw her in" の構文構造の作成に不要である.

本手法では, 新たな複数語の入力に対して, 操作 1), 2) を適用する前に, まず上昇型チャート解析を実行し, その後, 次の手続きにより不要な活性弧を取り除く.

不要な活性弧の除去

(I) 現在の入力の最後の単語にまたがる活性弧にチェックする.

Incremental Chart Parsing on a fragment-by-fragment basis:
Daisuke Mori, Yoshihide Kato, Shigeki Matsubara,
and Yasuyoshi Inagaki (Nagoya University)
daisuke@inagaki.nuie.nagoya-u.ac.jp

表 1: “Ken’s brother saw her in ” の入力に対する解析過程

入力語	チャート			
word	#	*loc*	*term*	選択チェック
Ken’s brother	5	0-2	$[[[Ken's]_{adj}[brother]_{n}]_{np}[?]_{vp}]_{s}$	√
saw	9	2-3	$[[saw]_{vt}[?]_{np}[?]_{pp}]_{vp}$	
her	15	2-4	$[[saw]_{vt}[[her]_{pron}]_{np}[?]_{pp}]_{vp}$	√
	17	3-4	$[[her]_{det}[?]_{n}]_{np}$	
in	19	5-6	$[[in]_{p}[?]_{np}]_{pp}$	√
	20	0-5	$[[[Ken's]_{adj}[brother]_{n}]_{np}[[saw]_{vt}[[her]_{pron}]_{np}[[?]_{pp}]_{vp}]_{s}$	
	21	0-6	$[[[Ken's]_{adj}[brother]_{n}]_{np}[[saw]_{vt}[[her]_{pron}]_{np}[[in]_{p}[?]_{np}]_{pp}]_{vp}]_{s}$	

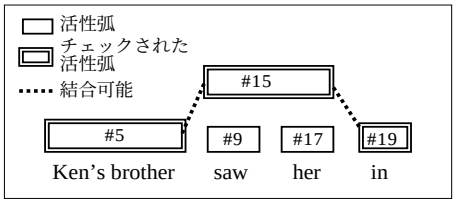

図 2: チェックされた活性弧

(II) チェックされた弧に，可能な限り操作 1) を適用する．作成された弧にチェックする．

(III) チェックされた弧と結合できる弧にチェックする．

(IV) チェックされていない弧を取り除く．

(I) 〜 (III) でチェックされなかった弧は，入力の最後の単語にまたがる弧と結合できないことを意味する．従って，これらの弧はそれまでの入力に対する構文構造を作成する上で不要である．それらを取り除くことにより，不要な弧に対する操作を避けることができ，解析の効率化につながる．

3.2 解析例

すでに入力された “Ken’s brother” の後で “saw her in ” が入力された場合について述べる．作成されるチャートの一部を表 1に示す．弧のチェック操作の結果を表 1の右端に示す．まず，上昇型チャート解析により，弧 #9 〜 #19 を生成する．不要な活性弧の除去操作により，まず，入力の最後の単語 “in” にまたがる弧 #19 にチェックし，この弧と結合できる弧 #15 に，続いて，弧 #15 と結合できる弧 #5 にチェックする．すなわち，図 2に示すようになる．活性弧 #9, #14, #17 はチェックされていないため，除去する．これらは “Ken’s brother saw her in” の構文構造の作成に不要な弧である．最後に，残された活性弧 “Ken’s brother” に対する弧 #5 に，弧 #15, #19 をそれぞれ順に結合し，弧 #20, #21 を得る．

4 実験と評価

本手法の有効性を評価するために， UltraSparcII ワークステーション (メモリ 512MB， CPU 248MHz) 上に， GNU Common Lisp 2.2.2 を用いて実装し，解析実験を行った．実験では，文献 [6] に収録された辞書 (438 語)，文法 (224 規則, 非終端記号数 93)，及び，英語文 (40 文，平均語数 12.2 語) を用いた．

複数語の長さを適切に決定することにより，効率のよい解析の実現が予想されるが，本実験では簡単のため，文頭から順に，2 単語，3 単語，4 単語，5 単語，6 単語ずつそれぞれ入力した．解析時間を，本手法と文献 [3] の手法の両方で測定した．入力の解析時間の各平均，及び，文献 [3] の手法に対する本手法の解析時間の比の平均を表 2に示す[1]．

表 2: 本手法と文献 [3] の手法の解析時間の比較

一回に入力する	平均解析時間 (秒)		文献 [3] に対する
単語数	本解析手法	文献 [3] の手法	比の平均 (%)
2 単語	0.63	4.57	55.9
3 単語	1.01	5.69	49.7
4 単語	1.23	6.36	41.5
5 単語	1.08	5.89	37.0
6 単語	1.20	8.84	33.8

文献 [3] の手法の場合，単語数が増えれば，削減される無駄な解析も増える． このため, 本手法を用いることにより，一度に入力される単語数が多くなるほど解析の効率も向上する．これにより本手法の有効性を確認した．

5 おわりに

本稿では，複数語ごとの入力に対する漸進的解析手法について述べた．本手法では，漸進的構文解析の一手法である漸進的チャート解析に，不要な構造を解析途中で取り除く操作を加えることにより，解析の効率化を実現している．また，解析実験による評価の結果，本手法の有効性を確認した．効率的な解析に適した複数語の長さを，解析途中で動的に決定する手法の検討は今後の課題である．

参考文献

[1] 秋葉, 田中：漸進的解析のための構文解析, 自然言語処理シンポジウム論文集, pp.39-48(1992).

[2] 加藤，松原，外山，稲垣：漸進的構文解析における構文的曖昧性とその解消，情報処理学会研究報告，NL-134，pp.117-122(1999).

[3] Matsubara, S. et al.: Chart-based Parsing and Transfer in Incremental Spoken Language Translation, *Proc. of NLPRS-97*, pp. 521–524 (1997).

[4] Steedman, M: Combinatory Grammar and Human Sentence Processing, *Modurality in Knowledge Representation and Natural Language Understanding*, pp. 187–205, MIT Press (1997).

[5] 田中 穂積：自然言語解析の基礎, 産業図書 (1989).

[6] Tomita, M: Efficient Parsing for Natural Language, *A Fast Algorithm for Practical Systems*, Kluwer Academic Publishers (1985).

[1] 文頭から複数語ずつ解析を行う場合，文の最後で単語が余る場合がある（例えば，11 単語からなる文を 3 単語ずつ解析するとき，文の最後で 2 単語だけ余る）．この場合，余った箇所は実験の評価対象から外した．