

PARSEVAL measures に基づく漸進的構文解析の定量的評価

加藤 芳秀^{†a)} (正員) 松原 茂樹^{††} (正員)

稲垣 康善^{†††*} (名誉員)

Quantitative Evaluation of Incremental Parsing by Extending the PARSEVAL Measures

Yoshihide KATO^{†a)}, Shigeki MATSUBARA^{††}, *Members*, and Yasuyoshi INAGAKI^{†††*}, *Fellow, Honorary Member*

[†] 名古屋大学大学院国際開発研究科, 名古屋市
Graduate School of International Development, Nagoya University, Nagoya-shi, 464-8601 Japan

^{††} 名古屋大学情報連携基盤センター, 名古屋市
Information Technology Center, Nagoya University, Nagoya-shi, 464-8601 Japan

^{†††} 愛知県立大学情報科学部, 愛知県
Faculty of Information Science and Technology, Aichi Prefectural University, Aichi-ken 480-1198 Japan

* 現在,

a) E-mail: yoshihide@gsid.nagoya-u.ac.jp

あらまし 構文解析の標準的な評価指標である PARSEVAL measures を拡張し, 漸進的構文解析を定量的に評価する手法を提案する. 解析途中で生成される構文解析結果と正解の部分構文木との一致の割合により漸進的構文解析の精度と漸進性を評価する.

キーワード 構文解析, PARSEVAL measures, 漸進性, ラベル精度, ラベル再現率

1. まえがき

漸進的構文解析とは, 文をその単語の出現順序に従って処理し, 文の入力途中の段階で, その構文構造を捉える枠組である. 同時通訳やリアルタイム字幕生成などの逐次的な言語処理システムの実現に必要な要素技術の一つである.

漸進的構文解析の性能評価においては, 最終的に得られる文全体に対する構文木のみならず, その解析過程において生成される中間解析結果, すなわち文の断片に対する部分構文木も評価の対象としなければならない. 逐次的な言語処理システムにおいて構文解析の中間解析結果が異なれば, その他のモジュールの動作 (例えば, 同時通訳システムにおける訳文生成など) にも影響を及ぼすからである.

ところが, そのような観点から漸進的構文解析を評価する方法については, これまで十分に検討されていない. 多くの文献, 例えば [2] では, 提案した漸進的構文解析を一文単位の構文解析と同様の方法により評価するのみで, 解析途中の振舞いについての評価は行われていない. 一方, 加藤らは, 漸進的構文解析を,

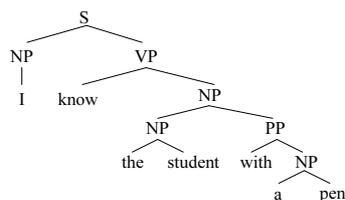


図 1 正解構文木
Fig. 1 Correct parse tree.

解析過程での解析精度と解析の漸進性の観点から評価している [3]. しかし, 正解構文木と完全に一致する中間解析結果のみを正解と判定するため, 解析結果が正解と部分的に一致している場合, それがどの程度一致しているかを評価できない. また, 解析の漸進性を解析処理全体の遅れにより評価しているが, 解析処理全体の遅れは漸進性に与える影響の一要因に過ぎず, 漸進性の評価としては不十分である. さらに, 精度と漸進性というトレードオフの関係にある 2 つの側面を統合的に評価することもできない.

本論文では, 理想的な漸進的構文解析が解析過程で明らかにする構文構造のうち, 実際の構文解析が再現できた割合をその構文解析の漸進性と定義し, 構文解析の精度と漸進性を評価する方法を提案する. 本手法は, 構文解析の標準的な評価指標である PARSEVAL measures [1] を拡張したものである. PARSEVAL では, 構文木を構成素に分解し, 解析結果の構成素のうち正解構成素と一致するものの割合により解析精度を評価する. 本手法ではこれを拡張し, 部分構文木中の構成素と正解構成素の一致に基づいて解析精度を評価する. これにより, 正解構文木との部分的な一致を考慮した精度評価が実現できる. また, PARSEVAL では, 構文解析により再現できた正解構成素の割合を構文解析の再現率と定義するが, 本手法では, 再現率を用いて構文解析の漸進性を評価する. 精度と再現率の総合的な評価指標である F 値 (精度と再現率の調和平均) により, 漸進的構文解析の精度と漸進性を総合的に評価することができる.

2. PARSEVAL measures

PARSEVAL measures [1] では, 構文木を構成素の集合として捕らえ, 正解構文木に対する構成素集合と構文解析結果に対する構成素集合の一致の割合により構文解析の性能を評価する. 以下では例として, 図 1 の構文木を正解構文木とし, 図 2 を構文解析結果として説明する.

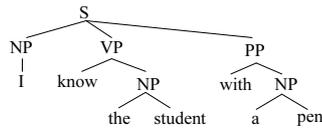


図2 構文解析結果
Fig. 2 Parse tree.

構成素は3項組 (s, t, X) として定義される。これは、文 $w_1 \dots w_n$ に含まれる部分単語列 $w_{s+1} \dots w_t$ ($0 \leq s < t \leq n$) が、範疇 X を構成することを意味する。図1の正解構文木に対する構成素集合は、 $\{(0, 1, NP), (1, 7, VP), (2, 7, NP), (2, 4, NP), (4, 7, PP), (5, 7, NP)\}$ であり、図2に対する構成素集合は、 $\{(0, 1, NP), (1, 4, VP), (2, 4, NP), (4, 7, PP), (5, 7, NP)\}$ である^(注1)。構文解析結果中の構成素のうち、正解と一致するものの割合をラベル精度と呼ぶ。正解構文木に対する構成素集合を T_C 、構文解析結果に対する構成素集合を T_G とするとき、ラベル精度は、 $P = |T_C \cap T_G| / |T_G|$ と定義される。図2の構文木中の構成素のうち、正解構成素に一致するのは、 $(0, 1, NP)$ 、 $(2, 4, NP)$ 、 $(4, 7, PP)$ 及び $(5, 7, NP)$ であり、ラベル精度は $4/5$ となる。

一方、ラベル再現率は、正解構文木中の構成素が構文解析によりどれだけ再現できたかを示す尺度であり、 $R = |T_C \cap T_G| / |T_C|$ と定義される。例において、ラベル再現率は $4/6$ となる。

ラベル精度とラベル再現率を統合的に評価する指標として、 F 値が用いられる。 F 値はラベル精度とラベル再現率の調和平均、すなわち $F = 2PR / (P + R)$ である。 F 値は0から1までの値をとり、ラベル精度、ラベル再現率がともに高い場合に高い値となり、どちらか一方でも低い場合は、低い値となる。

3. 漸進的構文解析の評価手法

漸進的構文解析は、文をその単語の出現順序に従って処理し、段階的に構文木を生成する。解析過程で生成される部分構文木は、途中まで読み込んだ文の断片に対する構文構造を表現している。漸進的構文解析を用いた逐次的言語処理システムにおいては、生成される部分構文木が正しいか否かが問題となるため、中間解析結果を対象とした精度評価が必要である。

さらに、解析の漸進性の観点、すなわち、解析途中の各段階において構文構造をどの程度明らかにしてい

(注1): 構文木の根に対応する構成素は開始記号であり常に一致するため、通常、これを取り除いて評価する。

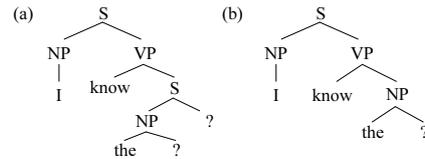


図3 “I know the” に対する部分構文木
Fig. 3 Partial parse trees for “I know the.”

るかの観点からの評価も重要である。例として、文の断片 “I know the” について考える。これに対する部分構文木としては、図3の(a)や(b)が考えられる。(a)は “know” が従属節をとり、“the” はその従属節の主語を構成する名詞句の先頭要素であるような部分構文木、(b)は “know” が目的語をとり、“the” がその目的語を構成する名詞句の先頭要素であるような部分構文木である。“I know the” に対する中間解析結果の選択方法としては、単純に(a)、(b)のいずれかを選択することが考えられるが、その一方で、別のアプローチが存在する。すなわち、先読み処理などにより解析処理を遅らせ、部分構文木の選択を延期する方法である。例えば、“I know the” に続く入力中に、従属節を構成する動詞が存在すれば、それを読んだ時点で(a)と決められる。このようなアプローチでは、解析精度の向上が期待できるものの、先読みした部分(例では、“student”)に関する構文構造を捉えることができず、解析の漸進性が損なわれることになる。このように、精度と漸進性の間にはトレードオフの関係があり、その両面から評価することが必要である。

また、漸進性に影響を与える別の要因として、解析戦略の違いが挙げられる。Lombardoらは、中間解析結果が構文構造を明示的に示す下降型構文解析を、上昇型構文解析や左隅型構文解析と比較して、漸進性が強い構文解析と定性的に位置づけているが[4]、このような解析戦略の違いに起因する漸進性の差は、解析処理の遅れによる評価では捉えることができない。

これらの問題を解決するために、本節では、2.で説明した PARSEVAL measures を拡張し、漸進的構文解析を評価する方法を提案する。

3.1 部分構文木の構成素

まず、漸進的構文解析において生成される部分構文木の構成素について考える。部分構文木は、文全体に対する構文木とは異なり、未完成の構成素が含まれていることが特徴である。構成素が完成しているかそうでないかを明示するために、本手法では構成素を4項

組 (s, t, X, c) に拡張し, その構成素が完成しているか否かを表現する. 4 番目の要素 c は, 0 か 1 のいずれかで, 前者がこの構成素が完成していないことを, 後者がこの構成素が完成していることを意味する. 例えば, 図 3 の部分構文木において, 記号 ? は構成素が未完成であること示しているが, 部分構文木 (a) に対する構成素集合は, $\{(0, 1, NP, 1), (1, 3, VP, 0), (2, 3, S, 0), (2, 3, NP, 0)\}$ である.

文 $w_1 \cdots w_n$ の断片 $w_1 \cdots w_i (i \leq n)$ に対する正解構成素集合 $T_C(i)$ は, 文全体に対する正解構成素集合 T_C をもとに次のように定義する.

$$\begin{aligned} T_C(i) &= \{(s, t, X, 1) \mid t \leq i \wedge (s, t, X) \in T_C\} \\ &\cup \{(s, i, X, 0) \mid s < i \wedge \exists t > i [(s, t, X) \in T_C]\} \end{aligned}$$

例えば, 図 1 を文全体の正解構文木とするとき, 文の断片 “I know the student” に対する正解構成素集合は, $\{(0, 1, NP, 1), (1, 4, VP, 0), (2, 4, NP, 0), (2, 4, NP, 1)\}$ である. $T_C(i)$ は, $w_1 \cdots w_i$ に対する構成素全体からなる集合であり, 理想的な漸進的構文解析であれば, これと完全に一致する部分構文木を解析の各段階において常に生成できる.

3.2 漸進的構文解析のラベル精度とラベル再現率

ラベル精度は, 解析結果中の構成素のうち, 正解構成素とマッチするものの割合であり, 部分構文木中の完成している構成素については, 従来通り正解構成素との完全なマッチングをとればよい. 一方, 未完成な構成素については, 部分的なマッチングも考慮したほうがよいと考えられる. 例として, 図 3 の部分構文木 (a) の構成素 $(2, 3, NP, 0)$ について考える. この構成素は正解構成素のいずれともマッチしないものの, “the” が NP の一部を構成しているという意味で, 正解構成素 $(2, 4, NP, 1)$ の一部分に相当すると捉えることができる. そこで本手法では, このような構成素も正解とマッチするものとみなす. 具体的には, 文の断片 $w_1 \cdots w_j$ に対する部分構文木 $T_G(j)$ の構成素と正解構成素とのマッチングを次のように定義する.

[定義 1] $x = (s, t, X, c) \in T_G(j)$, 及び $y = (s', t', X', c') \in T_C(i) (j \leq i)$ に対して, 次をすべて満たすとき, かつそのときに限り, $match(x, y)$ は成り立つ.

- $s = s'$.

入力: $T_G(j), T_C(i)$;
出力: $M(T_G(j), T_C(i))$;

$M(T_G(j), T_C(i)) = \phi$
for each $x \in T_G(j)$ do
 if there exists $y \in T_C(i)$ s.t. $match(x, y)$
 then delete a y from $T_C(i)$;
 add x to $M(T_G(j), T_C(i))$;

図 4 $M(T_G(j), T_C(i))$ の計算
Fig. 4 Computation of $M(T_G(j), T_C(i))$.

- $c = c' = 1, t = t'$ または, $c = 0, t \leq t'$ (注2).
- $X = X'$.

この定義を用いて, 正解構成素にマッチする部分構文木の構成素の集合 $M(T_G(j), T_C(i))$ を図 4 のように手続き的に定義する(注3).

以上を用いて, 漸進的構文解析のラベル精度 P を次のように定義する.

$$P = \frac{|M(T_G(j), T_C(i))|}{|T_G(j)|} \quad (1)$$

例として, 図 3 の部分構文木 (a) が “I know the student” に対する中間解析結果として生成された場合を考える. 定義 1 の意味で正解とマッチする構成素は, $(0, 1, NP, 1), (1, 3, VP, 0), (2, 3, NP, 0)$ であるので, ラベル精度は, $3/4$ となる.

次に再現率について考える. 正解構成素が部分構文木中の構成素と完全にマッチする場合, その構成素は完全に再現されている. 一方, 定義 1 の意味でマッチしている場合, それは正解構成素の一部分しか再現していない. 例えば, $(2, 3, NP, 0)$ は, 正解構成素 $(2, 4, NP, 1)$ の “the” に対応する部分のみを再現している. そこで本手法では, ラベル再現率の計算においては, 単純にマッチする構成素を数えるのではなく, 再現の割合により重み付けして計算する. 重み付けは, 構成素が覆っている範囲を用いて次のように定義する. 構成素 $x = (s, t, X, c)$ とそれにマッチする正解構成素 $y = (s', t', X', c')$ に対して,

$$w(x, y) = \frac{t - s}{t' - s} \quad (2)$$

と定義する. 完全に再現されている場合, すなわち $t = t'$ のとき, w は 1 となり, 一方, 再現される構成

(注2): $c = 0, c' = 1, t = t'$ のときもこの条件は満たされるが, これは, 構成素 x が空系列を補うことにより完成し, y と一致する場合があるためである.

(注3): $match(x, y)$ の意味で x とマッチする $y = (s', t', X', c') \in T_C(i)$ は複数存在する可能性があるが, 4. の実験では, t' が最小のものを選択している.

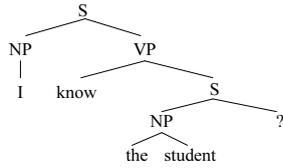


図5 “I know the student” に対する部分構文木
Fig.5 Partial parse tree for “I know the student.”

素の範囲が狭くなるにつれて、 w は小さくなる．この重み関数を用いて、漸進的構文解析のラベル再現率 R を次のように定義する．

$$R = \frac{\sum_{x \in M(T_G(j), T_C(i))} w(x, c(x))}{|T_C(i)|}$$

ただし、 $c(x)$ は、 $T_C(i)$ の要素で $match(x, c(x))$ が成り立つものとする．図 3(a) の例では、ラベル再現率は、 $(1 + \frac{2}{3} + \frac{1}{2})/4 = 13/24$ となる．

ここで定義したラベル再現率は、中間解析結果が正解構成要素をどの程度再現できたかを、言い換えれば、解析過程で構文構造をどの程度捉えることができたかを表現している．すなわち、ラベル再現率は、漸進性を測定する尺度とみなすことができる．

解析処理の遅れに起因する漸進性の低下について、例を用いて考える．1 語先読みを用いる場合、“I know the student” に対して、図 3(a) のような “I know the” に対する部分構文木が生成され、そのラベル再現率は上述のように $13/24$ である．一方、先読みを用いない場合、図 5 のような “I know the student” をすべて覆う部分構文木が生成される．この部分構文木のラベル再現率は $3/4$ である．図 3(a) の部分構文木のラベル再現率は、図 5 のそれと比べて低くなっている．このように、ラベル再現率により、解析の遅れに起因する漸進性の差を捉えることができる．

また、ラベル再現率により、構文解析戦略の違いに起因する漸進性の差も捉えることができる．例えば、図 5 の部分構文木は、下降型構文解析により生成される中間解析結果の一例であるが、これに対応する上昇型構文解析の中間解析結果は、この部分構文木から未完成な構成要素を取り除いたものである．すなわち、構成素 $(1, 4, VP, 0)$ や $(2, 4, S, 0)$ は認識されておらず、その構成素集合は、 $\{(0, 1, NP, 1), (2, 4, NP, 1)\}$ である．そのラベル再現率は、 $2/4$ である．下降型構文解析のラベル再現率は、上昇型構文解析のそれと比べて高くなっており、解析戦略の違いに起因する漸進性の

差を定量的に捉えることができる．

漸進性をラベル再現率により評価することにより、さらに、漸進性と精度を統合的に評価することができる．すなわち、ラベル精度とラベル再現率の F 値がその評価尺度である．

4. 評価手法の適用例

本節では、提案評価法の適用例として、上昇型構文解析、左隅型構文解析、下降型構文解析、及び、それらに先読みを導入した構文解析を評価する．

実験対象として、Penn Treebank [5] に収録されている構文木付き ATIS コーパス全 578 文を用いた．漸進的構文解析が対象とする文の断片は 4353 個である．構文解析には確率文脈自由文法を使用し、その文法規則は、578 文に付与された構文木から獲得した．文法規則の確率は最尤推定により計算した．

中間解析結果は、下降型構文解析については、文の断片に対する部分構文木の中で確率最大のものを選択した．左隅型構文解析、及び上昇型構文解析については、文の断片に対してそれを覆うような部分構文木は生成されず、確率の比較による選択はできないため、本実験では、それらの中間解析結果に相当するものを、下降型構文解析の中間解析結果をもとにして以下のように生成した．なお、以下では、文の断片 $w_1 \cdots w_i$ に対する下降型構文解析の中間解析結果を $TD(i)$ と記す．

上昇型構文解析の中間解析結果 $TD(i)$ 中のすべての節点 τ について、 τ が未完成な（子孫に？が出現する）節点であるならば、 τ を取り除く．

左隅型構文解析の中間解析結果 $TD(i)$ 中のすべての節点 τ について、 τ の左端の子が未完成であるならば、 τ を取り除く．

先読みを用いた構文解析についてもその中間解析結果を、下降型構文解析の中間解析結果をもとに生成した．以下がその手順である．

k 語先読み $TD(i)$ 中のすべての節点 τ について、 τ の左端の単語が $w_l (i - k < l)$ であるならば、 τ を取り除く．

この手順により生成される中間解析結果は、 $w_1 \cdots w_{i-k}$ に対する部分構文木となる．すなわち、 w_{i-k+1} 以降の構造はすべて未完成として扱われる．

下降型、左隅型、上昇型、及びそれらに 0~2 語先読みを導入した構文解析について、ラベル精度と漸進性を測定した．評価結果を表 1 に示す．

処理の遅れにより漸進性を評価する場合、下降型、

表 1 評価結果
Table 1 Experimental results.

種類	先読み (語)	再現率 (%)	精度 (%)	F 値
下降型	0	86.7	92.1	89.3
	1	65.2	92.5	76.5
	2	51.6	92.6	66.3
左隅型	0	78.2	93.7	85.2
	1	60.4	93.8	73.5
	2	48.2	93.8	63.7
上昇型	0	42.0	92.3	57.8
	1	41.8	92.2	57.5
	2	35.5	91.9	51.2

左隅型, 上昇型構文解析の漸進性は, 先読み語の数として評価することになる. しかし, 先読み語の数と同じであっても, 再現率は, 下降型, 左隅型, 上昇型の順に大きく減少する. このことは, 解析処理のタイミングが同じであっても, 中間解析結果の表現する構文構造は大きく異なることを意味する. 逆に言えば, 処理の遅れによる漸進性の評価は, 処理タイミングが早いのみで, 文の断片の構文構造を明らかにできないような構文解析を, 漸進性が高い解析と評価する危険性があることを意味する. 再現率による漸進性の評価を用いれば, この問題は回避できる. 一方, 先読みにより解析処理が遅れるにつれて, 再現率も減少していることから, 再現率は, 処理の遅れについても反映した漸進性の評価尺度であると位置づけることができる.

5. あとがき

本論文では, 漸進的構文解析の精度と漸進性を定量的に評価する指標を提案し, 構文解析実験により, その有効性を確認した.

漸進的構文解析が用いられるアプリケーションによっては, 精度, あるいは漸進性のどちらか一方が他方に優先されるケースが考えられる. 精度と漸進性に重み付けを行い F 値を求めることにより, 一方に重点を置いた評価ができるが, アプリケーションに依存した評価についても今後検討したい.

謝辞 本論文を改善する上で, 査読者の方には貴重なご意見を頂きました. ここに深く感謝いたします. 本研究の一部は, 科学研究費若手研究 (B)(課題番号: 17700145) の助成を受けています.

文 献

- English grammars”, Proc. Speech and Natural Language Workshop, pp.306–311, 1991.
- [2] M. Collins and B. Roark, “Incremental parsing with the perceptron algorithm”, Proc. 42nd Annual Meeting of the Assoc. for Computational Linguistics, pp.112–119, 2004.
- [3] Y. Kato, S. Matusbara and Y Inagaki, “Stochastically evaluating the validity of partial parse trees in incremental parsing, Proc. of the ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together, pp. 9–15, 2004.
- [4] V. Lombardo and P. Sturt, “Incremental Processing and Infinite Local Ambiguity”, Proc. of the 19th Annual Conference of the Cognitive Science Society, pp.448–453, 1997.
- [5] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, “Building a large annotated corpus of English: the Penn Treebank”, Computational Linguistics, vol.19, no.2, pp.310–330, 1993.
- (平成 xx 年 xx 月 xx 日受付)
- [1] E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini and T. Strzalkowski, “A procedure for quantitatively comparing the syntactic coverage of

Abstract This paper proposes a method for evaluating incremental parsers quantitatively. The method defines two measures of incremental parser performance: precision and incrementality by extending the PARSEVAL measures, which are well-known measures for phrase-structure parsers.

Key words parsing, PARSEVAL measures, incrementality, labelled precision, labelled recall