誤り周辺の文脈を考慮した構文木コーパスの自動訂正手法

鈴木 寛大 † 加藤 芳秀 ‡ 松原 茂樹 †
† 名古屋大学大学院情報科学研究科
‡ 名古屋大学情報連携統括本部
ksuzuki@db.ss.is.nagoya-u.ac.jp

1 はじめに

近年，様々なタグ付きコーパスが開発され，自然言語処理の研究に利用されている．一般に，タグ付け作業には人手による作業が介入するが，それが原因でタグに誤りが混入し，コーパスの質の低下といった問題が生じる．これに対して，さまざまなタグ付きコーパスについて，それに含まれる誤りを検出・訂正する研究が行われている [1]．Kato ら [3] 及び Suzuki ら [5] は構文木コーパスに含まれる誤りを訂正する手法を提案している．これらの手法は，誤った木構造を正しい木構造に変換するルールを抽出し，それをコーパスに適用することで誤り訂正を実現する．抽出したルールの中には，同一の木構造をそれぞれ別の木構造に変換する複数のルールが存在する場合がある．このとき，これらの手法は複数のルールのうち，いずれか一つをコーパス中の誤り候補に一律に適用する．しかし，たとえ誤りの構造が同一であっても，出現箇所によって適用すべきルールが異なる場合がある．また，抽出したルールの変換元の木構造が必ずしも誤りであるとは限らない．そのため，ルールの変換元の木構造の出現箇所ごとに，その構造が誤りであるのか，また，誤りであれば適切なルールがどれであるかを判断する必要がある．

本稿では，誤り候補の出現箇所に応じて適切な誤り訂正ルールを選択する手法を提案する．本手法では，誤り候補の出現箇所一つ一つに対して誤り訂正ルールの適用に関する評価尺度を定義する．この評価尺度は，誤り候補の周辺の文脈を条件とした場合の訂正先の木構造の出現確率として定義される．実際のコーパスを用いた実験によって，誤り候補の周辺の文脈を考慮することが適切な誤り訂正ルールの選択に有用であることが確認できた．

2 関連研究

本節では，構文木コーパスの誤り訂正における Kato らの手法 [3] と Suzuki らの手法 [5] を説明し，その問題点を議論する．これらの手法はいずれも synchronous

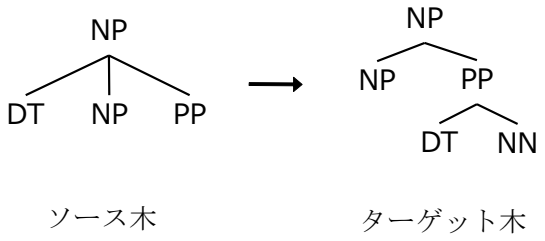


図 1: 誤り訂正ルール

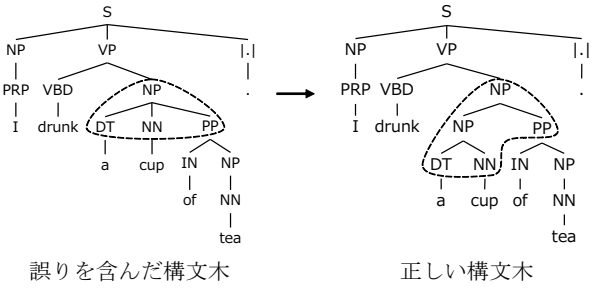


図 2: 誤り訂正ルールの適用

tree substitution grammar (STSG)[2] に基づく誤り訂正ルールをコーパスから抽出する．ルールは基本木と呼ばれる木のペアで構成され，一方の木はソース木，他方の木はターゲット木と呼ばれる．誤り訂正ルールの例を図 1 に示す．両基本木の根節点や葉節点は一対一の対応が取られており，根節点のラベル，及び葉節点のラベルの系列は一致する必要がある．ルールは，構文木中のソース木にマッチする木構造をターゲット木に変換する．図 1 に示したルールの適用例を図 2 に示す．点線部はソース木及びターゲット木にマッチする木構造である．ルールの抽出方法は異なるが，いずれの手法も誤り訂正の実現方法は同一であり，コーパス中の構文木に含まれる誤った木構造をソース木に，その箇所に対する訂正結果に相当する木構造をターゲット木に持つルールを抽出し，コーパスに適用する．また，抽出したルールの中から誤り訂正に適切なものを選択するために，ルールのスコアを定義している．

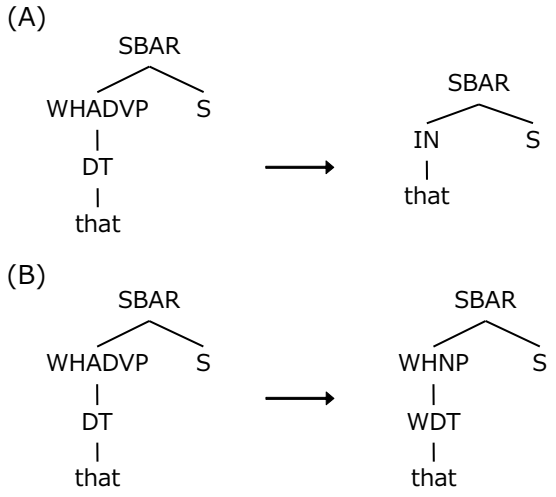


図 3: 同一のソース木を持つ誤り訂正ルール

ソース木 s 及びターゲット木 t から成るルール $\langle s,t\rangle$ のスコアは以下のように定義される．

$$Score(\langle s,t\rangle) = \frac{f(t)}{f(s)+f(t)} \quad (1)$$

ここで，$f(\cdot)$ はコーパス中の基本木の出現頻度を表す．$f(s)$ が $f(t)$ に比べて小さいほど，$Score(\langle s,t\rangle)$ は大きくなり，$\langle s,t\rangle$ が誤り訂正に適切であるとみなされる．

コーパスから抽出したルールの中には，同一のソース木を持つルールが複数存在する場合がある．このとき，従来手法ではそれらのルールのうち，スコアが最大となるルールが採用される．ソース木が同一であるため，ターゲット木の出現回数が最も多いルールのスコアが最大になり，そのルールがコーパスに適用される．しかし，このようなルールがコーパスに出現する誤りを適切に訂正するとは限らない．例えば，同一のソース木を持つ図 3 のようなルールが抽出されたと仮定する．さらに，コーパス中に誤り訂正ルール (A),(B) のソース木を含んだ構文木 (a),(b)(図 4) が存在する場合を考える．点線部はルールのソース木にマッチする木構造である．(a) に対しては (A) を，(b) に対しては (B) を適用するのが適切であるが，従来の手法ではどちらの構文木にもターゲット木の出現回数が多い方のルールが適用される．スコアに基づき，ルール (A) が採用されれば (a)，ルール (B) が採用されれば (b) は誤りが訂正されるが，他方の構文木に含まれる誤りは訂正されない．すなわち，式 (1) のスコアを用いてルールを選択した場合，(A) と (B) に含まれる誤りを共に訂正することは原理的に不可能である．

また，誤り訂正ルールのソース木が必ずしも誤りとは限らない．誤りでないソース木を変換することは，逆に誤りを生むことを意味するため，これを防ぐ必要がある．

鈴木らはソース木の出現箇所ごとに適用すべきルールを選択する手法 [6] を提案している．この手法は，訂正対象の構文木中のソース木の葉に支配された木構造とターゲット木を含む構文木中の対応する箇所の木構造がどれだけ類似するかを共通する部分木の数で評価し，適用するルールを選択する．しかし，ソース木の根の上部に存在する木構造については一切考慮していない．

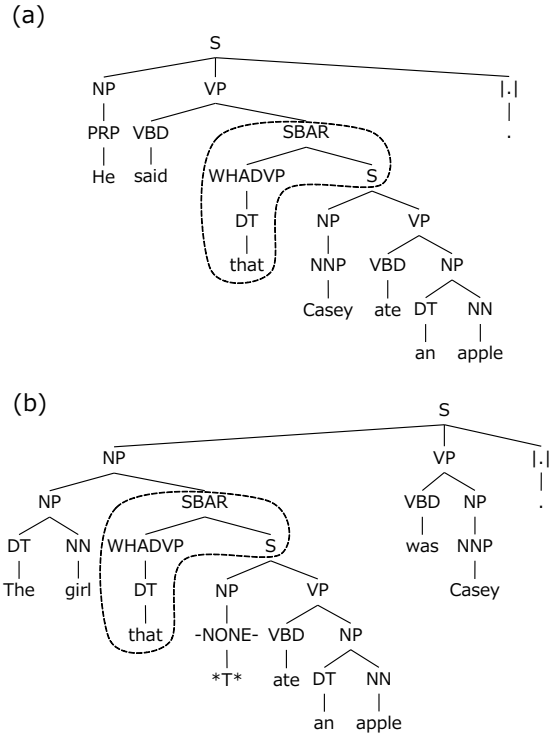


図 4: ルール (A),(B) のソース木を含む構文木

3 周辺文脈を考慮した誤り訂正ルールの選択

本節では，前節で述べた問題を解決するために，コーパス中の構文木に出現したソース木に対して適切な誤り訂正ルールを選択する手法を提案する．提案手法ではソース木の出現箇所に応じてルールの適用に関する評価尺度を定義する．評価尺度は構文木中のソース木周辺の文脈を条件としたターゲット木の出現確率として定義される．ソース木の出現箇所ごとに各ルールに対する評価尺度の値を求め，最大の値をとるルールを適用する．

まず，基本的なアイディアについて述べる．構文木に誤り訂正ルールのソース木が含まれており，それが誤りだとしても，その周辺の文脈には誤りは含まれないと仮定する．このとき，ソース木に対応するターゲット木が，同じ文脈を周囲に持つという条件において出現する確率が高いほど，誤り訂正結果として適切であ

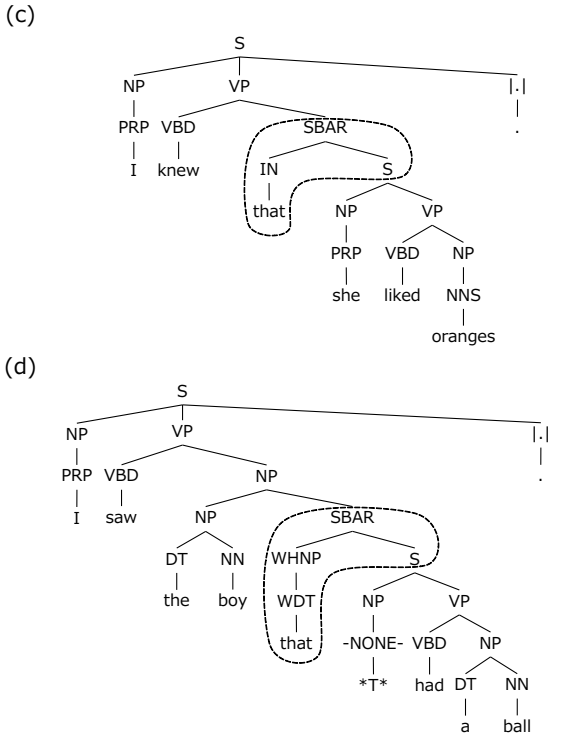


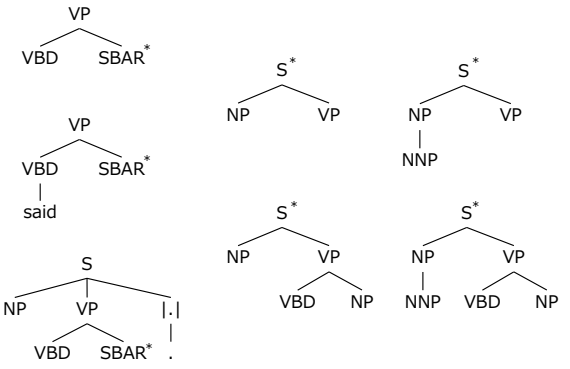
図 5: ルール (A),(B) のターゲット木を含む構文木

ると考えられる．

例として，コーパスが図 4 の構文木 (a),(b) に加え，構文木 (c),(d)(図 5) を含む場合を考える．(c) にはルール (A) のターゲット木が，(d) にはルール (B) のターゲット木が出現する．点線部は各ルールのターゲット木にマッチする部分を示している．ここで，図 4 の構文木 (b) に出現するソース木に対して適切なルールがいずれであるかを考える．構文木 (b) では，ソース木の葉節点 S の子孫として wh 移動の痕跡*T*が出現する．ターゲット木の葉節点 S の子孫に*T*が出現するという条件で，そのターゲット木がルール (A),(B) のどちらのターゲット木であるかを考えた時，ルール (B) のターゲット木である確率が高い．なぜなら，*T*の前方にルール (A) のターゲット木に含まれる接続詞の ‘that’ が出現することは稀であるが，ルール (B) のターゲット木に含まれる主格の関係代名詞 ‘that’ は頻出するためである．そのため，構文木 (b) に出現するソース木にはルール (B) を適用することが適切である可能性が高いと言える．このようにソース木周辺の文脈を条件にした時の各ターゲット木の出現確率を考慮することで，適切なルールを選択できると考えられる．

3.1 基本木の周辺部分木

評価尺度の説明に入る前に，必要な定義を行う．以下では，部分木とは「構文木中の構文規則を保存した連結部分グラフ」を指すものとする．また，構文木 σ



ソース木の根の上部に出現する部分木の一部　　ソース木の葉の下部に出現する部分木の一部

図 6: 構文木 (a) のソース木周辺部分木の例

に出現する基本木 e の周辺部分木とは，以下のいずれかの条件を満たすような σ の部分木 m を指すものとする．

(1) m の葉節点の一つは e の根節点である．

(2) m の根節点は e のいずれかの葉節点である．

条件 (1) を満たす部分木は基本木の根節点の上部に出現し，条件 (2) を満たす部分木は基本木の葉節点の下部に出現する．例として，図 4 中の構文木 (a) のソース木周辺の部分木の一部を図 6 に示す．部分木中の $*$ がついた節点はソース木の根もしくは葉に相当する節点である．

3.2 誤り周辺の文脈を考慮した評価尺度

本節では，コーパス中の構文木 σ において節点 η を根として出現するソース木 s に対してルール $\langle s,t\rangle$ を適用することに関する評価尺度 $Score(\sigma,\eta,\langle s,t\rangle)$ を定義する．

基本木 s をソース木とするルールが n 個存在し，それぞれのターゲット木を $t_1,t_2,\ldots,t_n$ とする．ソース木 s 自身もターゲット木に含めるものとする．このルールはソース木にマッチした部分を誤りとみなさないルールである．ソース木 s を含んだ構文木 σ には，s の周辺部分木が m 個存在し，それぞれを $\tau_1,\tau_2,\ldots,\tau_m$ とする．σ,η,s が定まると，$\tau_1,\tau_2,\ldots,\tau_m$ も定まる．この時，評価尺度 $Score(\sigma,\eta,\langle s,t\rangle)$ を以下のように定義する．

$$Score(\sigma,\eta,\langle s,t\rangle) = P(t|\tau_1,\tau_2,\ldots,\tau_m) \qquad (2)$$

条件付確率 $P(t|\tau_1,\tau_2,\ldots,\tau_m)$ は周辺部分木として $\tau_1,\tau_2,\ldots,\tau_m$ が与えられた場合の t の出現確率であ

る．本手法ではこれを以下のように近似する．

$$P(t|\tau_1, \tau_2, \ldots, \tau_m) \approx \alpha_0 P(t) + \sum_{i=1}^{m} \alpha_i P(t|\tau_i) \quad (3)$$

ただし，$\sum_{i=0}^{m} \alpha_i = 1$ である．つまり，ソース木の周辺部分木それぞれで確率モデルを求め，その混合モデルとして定義する．式 (3) の右辺の各項は以下のように求める．

$$P(t) = \frac{f(t)}{\sum_{i=1}^{n} f(t_i)} \quad (4)$$

$$P(t|\tau) = \frac{f(combine(t, \tau))}{\sum_{i=1}^{n} f(combine(t_i, \tau))} \quad (5)$$

ここで，$combine(t, \tau)$ は，ターゲット木 t にソース木 s の周辺部分木 τ を連結させた部分木である．τ が s の根節点の上部に出現する場合は，t の根節点に τ を連結させ，τ が s の葉節点の下部に出現する場合は，t の対応した葉節点に τ を連結させる．式 (4) 及び式 (5) を計算する際には，現在ルール適用の対象となっている s の出現箇所については考慮しないものとする．

4 実験

提案手法の有用性を確認するために，構文木コーパスである Penn Treebank[4] を用いて実験を行った．

まず，Penn Treebank の Wall Street Jarnal コーパスの全 49208 文を対象に，Suzuki らの手法 [5] を用いて誤り訂正ルールを抽出した．次に，抽出した全 2379 個のルールを従来手法のスコア (式 (1)) をキーにソートした．スコア上位のルールのソース木から順に，そのソース木の各出現箇所に適用するルールを提案手法を用いて選択した．ソース木の周辺部分木の総数が n 個であった場合，式 (3) の右辺の各係数は全て均一に $\frac{1}{n+1}$ とした．いずれかの箇所で従来手法と提案手法では異なるルールが適用されたソース木上位 50 個に対し，異なるルールが適用された箇所で選択したルールが適切であるかを判定した．判定した箇所の総数は 88 箇所である．選択したルールが適切であったかどうかの内訳を表 1 に示す．提案手法が適切なルールを選択した 43 箇所について，選択したルールの内訳を表 2 に示す．複数のルールが適用できるソース木に対して適切なルールが選択できることに加え，ソース木が誤りであるかどうかの判断に成功していることが確認できた．

表 1: 選択ルール適用の判定結果

提案手法が選択したルールが適切	43
従来手法が選択したルールが適切	13
どちらの判断も不適切	32

表 2: 提案手法が正しく選択したルールの内訳

ターゲット木 $\neq$ ソース木	7
ターゲット木 $=$ ソース木	36

5 おわりに

本稿では，構文木コーパスに出現する誤りに対して，複数の誤り訂正ルールが適用できる場合に，どのルールが適切かを判断する手法を提案した．本手法では，ソース木の周辺の文脈を条件としたターゲット木の出現確率をルール適用に関する評価尺度として定義した．この評価尺度を用いて，ソース木の出現箇所ごとに，適切な誤り訂正ルールがどれであるかを判断する．実験により，適切な訂正候補の選択において，誤りの周辺の文脈を考慮することの有用性が確認できた．

参考文献

[1] Markus Dickinson. Detection of annotation errors in corpora. *Language and Linguistics Compass*, Vol. 9, No. 3, pp. 119–138, 2015.

[2] Jason Eisner. Learning non-isomorphic tree mappings for machine translation. In *Proc. 41st Annual Meeting on Association for Computational Linguistics*, pp. 205–208, 2003.

[3] Yoshihide Kato and Shigeki Matsubara. Correcting syntactic annotation errors using a synchronous tree substitution grammar. *IEICE Transactions on Information and Systems*, Vol. E93-D, No. 9, pp. 2660–2663, 2010.

[4] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.

[5] Kanta Suzuki, Yoshihide Kato, and Shigeki Matsubara. Correcting errors in a treebank based on tree mining. In *Proc. 10th International Conference on Language Resources and Evaluation*, pp. 1540–1545, 2016.

[6] 鈴木寛大, 加藤芳秀, 松原茂樹. 構文的類似度を考慮した構文木コーパスの誤り訂正. 言語処理学会第 22 回年次大会発表論文集, pp. 166–169, 2016.