対訳コーパスを利用した構文解析器の自己学習

森下 睦　　赤部 晃一　　波多腰 優斗 *
Graham Neubig　　吉野 幸一郎　　中村 哲
奈良先端科学技術大学院大学 情報科学研究科
{morishita.makoto.mb1, akabe.koichi.zx8,
neubig, koichiro, s-nakamura}@is.naist.jp
hatakoshi.yuto@gmail.com

1 はじめに

統計的手法を用いた構文解析器では，学習に用いる構文木データの量が解析精度に大きく影響する．また，構文解析器の学習データが網羅していない分野に関する文については，解析精度が低くなる傾向があり，これを解決するために様々な分野のデータが必要とされている [2]．しかし，構文木データを作成するためには人手による構文木のアノテーション作業が必要となるため，大規模かつ様々な分野のデータを作成するためには大きなコストがかかってしまう．

このような現状において，構文解析器の精度を高める手法の一つとして自己学習 (Self-Training) が挙げられる [9]．構文解析器の自己学習とは，既存の構文木データで学習した構文解析器に，新たな文を入力し構文木を自動生成し，得られた構文木を用いて再度モデルの学習を行う手法である．これにより，追加のアノテーションを必要とせずに学習データが増え，構文解析精度が向上する．しかしこの手法の問題点として，自動生成した構文木は必ずしも正しくなく，誤った構文木が学習データに混入することで，自己学習の効果が低下する点が挙げられる．これを解決するために，Katz-Brown らは自己学習に用いるデータを選択する標的自己学習 (Targeted Self-Training) を提案している [5]．この手法では，出力された複数の構文木候補を基に，機械翻訳の前処理である事前並べ替えを行い，並び替え結果と正解並べ替えデータを比較し，最も良い結果で使われた構文木を自己学習に利用する．これにより，複数の構文木候補から正しい構文木を選択することができ，学習データに誤った構文木が混入しにくくなり，自己学習の効果が大きくなる．しかし，この手法では人手で正解並び替えデータを用意する必要があり，これにコストがかかってしまうため大規模なデータを対象とすることは難しい．

本稿では，この問題を対訳コーパス，統語ベース翻訳器，機械翻訳の自動評価尺度を用いて，自己学習データを選択する手法を提案する．本手法では追加の正解データを必要とせず，既存の対訳コーパスのみを用いて構文解析器の精度を向上できる利点がある．本手法を用いて，科学論文を抜粋した対訳コーパス (ASPEC) に対して標的自己学習を行った結果，構文解析精度がベースラインおよび従来の自己学習手法で自己学習した場合と比較して有意に向上した [1]．

2 構文解析の自己学習

2.1 自己学習の概要

構文解析器の自己学習とは，既存のモデルで学習した構文解析器が解析・生成した構文木を学習データとして用いることで，構文解析器を再学習する手法である．言い換えると，自己学習対象の各文 $\boldsymbol{f}$ に対して，式 (1) に基づいて確率が最も高い構文木 $\hat{T}_{\boldsymbol{f}}$ を求め，この構文木を構文解析器の再学習に用いる．この手法は追加のアノテーションを必要としないため，構文解析器の学習データ量が大幅に増え，解析精度が向上する．

$$\hat{T}_{\boldsymbol{f}} = \underset{T_{\boldsymbol{f}}}{\operatorname{argmax}}\, Pr(T_{\boldsymbol{f}}|\boldsymbol{f}) \tag{1}$$

Charniak は，Wall Street Journal (WSJ) コーパス [8] によって学習された確率文脈自由文法 (Probabilistic Context-Free Grammar, PCFG) モデルを用いた構文解析器では，自己学習の効果は得られなかったと報告している [1]．一方で，潜在クラスを用いることで構文解析の精度を向上させた PCFG-LA (PCFG with Latent Annotations) モデルは自己学習により大幅に解析精度が向上することが知られている [4]．これは，PCFG-LA モデルを用いることで自動生成された構文木の精度が比較的高くなることに加え，PCFG-LA モデルが通常の PCFG モデルと比べて多くのパラメータを持つので，学習データが増加する恩恵が大きいことが理由として挙げられる．これらの先行研究を基にして，本稿では PCFG-LA モデルを用いた構文解析器の自己学習を考える．

2.2 標的自己学習

従来の自己学習手法では，構文解析器が生成した構文木全てを自己学習に用いていたため，構文解析誤りが自己学習に悪影響を与えていた．この問題を解決するために，全学習データの中から精度向上に寄与するものだけを選択する「標的自己学習」が提案されている．

Katz-Brown ら [5] は構文解析器の自己学習をフレー

*現在，セイコーエプソン株式会社

[1] 本稿の詳細は [13] を参照

ズベース翻訳のための事前並べ替えに適用する手法を提案している．フレーズベース翻訳のための事前並べ替えとは，原言語文の単語を目的言語の語順に近くなるように並び替えることによって，機械翻訳の精度を向上させる手法である．この手法では，構文解析器を用いて一文から複数の構文木候補を出力し，この構文木候補を用いて事前並べ替えを行う．その後，並べ替え結果を人手で作成された正解並べ替えデータと比較することによって，各出力にスコアを割り振る．これらの並び替え結果のスコアを基に，構文木候補の中から最も高いスコアを獲得した構文木を選択し，この構文木を自己学習に使用する．このように，学習に用いるデータを選択し，自己学習を行う手法を標的自己学習 (Targeted Self-Training) という．Katz-Brown らの手法では，正解並べ替えデータを用いて，自己学習に使用する構文木を選択することで，誤った並べ替えを行う構文木を取り除くことができ，学習データのノイズを減らすことができる．しかし，この手法を適用するためには人手で作成された正解並べ替えデータが必要となり，このデータを作成するために大きなコストがかかってしまうという欠点がある．

本稿では，統語ベース翻訳器を使って対訳コーパスに含まれる文を翻訳し，翻訳文を機械翻訳の自動評価尺度を使って評価し，自己学習に使用するデータを選択する手法を提案する．本手法では，対訳以外に新たな正解データを作成する必要が無く，既存の対訳コーパスを構文解析器の精度向上に利用することができる利点がある．

以降では，本手法で必要となる統語ベース翻訳の一種である Tree-to-String 翻訳について説明する．

3 Tree-to-String 翻訳

統計的機械翻訳では，原言語文 $\boldsymbol{f}$ が与えられた時に，目的言語文 $\boldsymbol{e}$ へと翻訳される確率 $Pr(\boldsymbol{e}|\boldsymbol{f})$ を最大化する $\hat{\boldsymbol{e}}$ を推定する問題を考える．

$$\hat{\boldsymbol{e}} \coloneqq \underset{\boldsymbol{e}}{\operatorname{argmax}} Pr(\boldsymbol{e}|\boldsymbol{f}) \tag{2}$$

様々な手法が提案されている統計的機械翻訳の中でも，Tree-to-String (T2S) 翻訳は原言語文の構文木 $T_{\boldsymbol{f}}$ を使用することで，原言語文に対する解釈の曖昧さを低減し，原言語と目的言語の文法上の関係をルールとして表現することで，より精度の高い翻訳を実現する．T2S 翻訳は下記のように定式化される．

$$\begin{aligned}
\hat{\boldsymbol{e}} &\coloneqq \underset{\boldsymbol{e}}{\operatorname{argmax}} Pr(\boldsymbol{e}|\boldsymbol{f}) && (3)\\
&= \underset{\boldsymbol{e}}{\operatorname{argmax}} \sum_{T_{\boldsymbol{f}}} Pr(\boldsymbol{e}|\boldsymbol{f}, T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}) && (4)\\
&\simeq \underset{\boldsymbol{e}}{\operatorname{argmax}} \sum_{T_{\boldsymbol{f}}} Pr(\boldsymbol{e}|T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}) && (5)\\
&\simeq \underset{\boldsymbol{e}}{\operatorname{argmax}} Pr(\boldsymbol{e}|\hat{T}_{\boldsymbol{f}}) && (6)
\end{aligned}$$

ただし，$\hat{T}_{\boldsymbol{f}}$ は構文木の候補の中で，最も確率が高い構文木であり，式 (1) で表される．

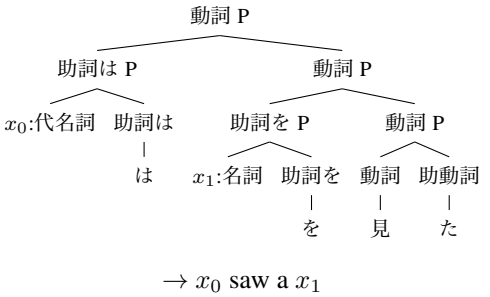


図 1: 日英 T2S 翻訳における翻訳ルールの例

図 1 に示すように，T2S 翻訳[2] によって用いられる翻訳ルールは，置き換え可能な変数を含む原言語文構文木の部分木と，目的言語文単語列の組で構成される．図 1 の例では，x_0，x_1 が置き換え可能な変数である．これらの変数には，他のルールを適用することにより翻訳結果が挿入され，変数を含まない出力文となる．訳出の際は，翻訳ルール自体の適用確率や言語モデル，その他の特徴などを考慮して最も事後確率が高い翻訳結果を求める．また，確率の高い n 個の翻訳結果を出力することもでき，これを n-best 訳という．

T2S 翻訳では，原言語文の構文木を考慮することで，語順が大きく異なる言語対の翻訳がフレーズベース翻訳と比べて正確になる場合が多い．しかし T2S 翻訳は，構文木を翻訳に利用するため，翻訳精度が構文解析器の精度に大きく依存するという欠点がある．この欠点を改善するために，複数の構文木を構文森と呼ばれる超グラフ (Hyper-Graph) の構造で保持し，構文森を翻訳に使用する Forest-to-String (F2S) 翻訳 [11] も提案されている．構文森を翻訳に用いることで，翻訳器は複数ある構文木の候補から構文木を選択することができ，翻訳精度の改善に繋がる [18]．F2S 翻訳は下記のように定式化される．

$$\langle \hat{\boldsymbol{e}}, \hat{T}_{\boldsymbol{f}} \rangle = \underset{\langle \boldsymbol{e}, T_{\boldsymbol{f}} \rangle}{\operatorname{argmax}} Pr(\boldsymbol{e}|T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}) \tag{7}$$

4 対訳コーパスを利用した構文解析器の標的自己学習

本稿では，対訳コーパス，統語ベース翻訳および機械翻訳の評価尺度を利用し，使用するデータを選択した上で，構文解析器の自己学習を行う．提案手法の概要を図 2 に示す．図のように原言語文を構文解析器に入力し，出力された構文森を F2S 翻訳器に入力する．これにより n-best 訳と，翻訳に使われた構文木のペアが出力される．その後，参照訳と機械翻訳の自動評価尺度を用いて，n-best 訳に対して翻訳精度のスコア付けを行う．ここで得られたスコアを基に学習データを選択し，自己学習を行う．

データの選択には，構文木の選択法および文の選択法を組み合わせて使用する．構文木の選択法では，一つの文の構文木候補から誤りの少ない構文木を選択し，

[2] 具体的には，木トランスデューサ (Tree Transducers) を用いた T2S 翻訳．

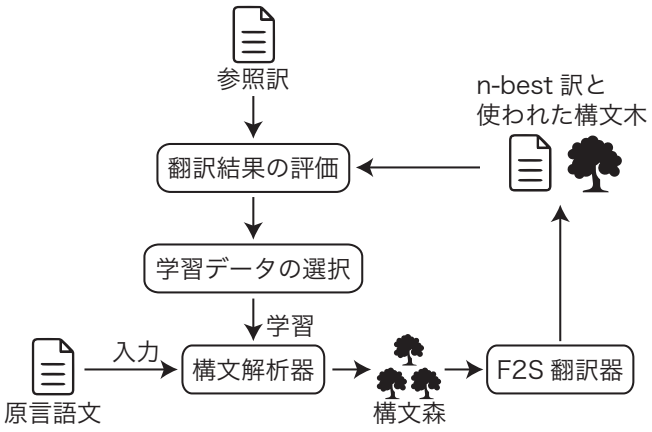


図 2: 提案手法の概要

文の選択法では，コーパス全体から精度向上に有効な文のみを選択する．以降ではそれぞれの手法について説明する．

4.1 構文木の選択法

翻訳の際，翻訳器は複数の翻訳候補の中から，最も翻訳確率が高い訳を 1-best 訳として出力する．しかし，実際には翻訳候補である n-best 訳の方が参照訳に近く，翻訳器が出力した 1-best 訳よりも翻訳精度が高いと思われる場合が存在する．そこで本稿では，翻訳候補 $\boldsymbol{E}$ の中から最も参照訳 $\boldsymbol{e}^*$ に近い訳を Oracle 訳 $\bar{\boldsymbol{e}}$ と定義し，Oracle 訳 $\bar{\boldsymbol{e}}$ に使われた構文木を自己学習に使用する．翻訳候補 $\boldsymbol{e}$ と参照訳 $\boldsymbol{e}^*$ の類似度を表す評価関数 $\mathrm{score}(\cdot)$ を用いて，Oracle 訳 $\bar{\boldsymbol{e}}$ は下記の通り表される．

$$\bar{\boldsymbol{e}} = \underset{\boldsymbol{e} \in \boldsymbol{E}}{\mathrm{argmax}}\, \mathrm{score}(\boldsymbol{e}^*, \boldsymbol{e}) \tag{8}$$

4.2 文の選択法

4.1 節では，1 つの対訳文の n-best 訳から誤りの少ない構文木を選択する方法について述べた．しかし，正しい訳が n-best 訳の中に含まれていない場合も多くあり，これらの文は学習に用いることそのものが構文解析器の精度低下を招く可能性がある．そのため，n-best 訳の中に良い訳が含まれていない場合その文を削除するように，学習データ全体から自己学習に用いる文を選択する手法を提案する．

F2S 翻訳では，正しく翻訳するためには正しい構文木が必要となる．このため，翻訳文の自動評価値が低い場合，翻訳時に正しい構文木が使われていない可能性が高く，これらの構文木を使うと自己学習のノイズとなる可能性が高い．そこで，自動評価値が低いデータを学習データから取り除くことで，学習データ中のノイズが減り，より正確な構文木のみが残ると考えられる．本手法では，Oracle 訳の自動評価値が上位の文に使用された構文木を自己学習に使用する．

5 実験評価

5.1 実験設定

本研究では，日本語の構文解析器を用いる日英翻訳器を利用して実験を行った．翻訳データおよび自己学習用データとして，科学論文を抜粋した対訳コーパスである ASPEC[3] を使用した．構文解析器には [15] で最も高い日英翻訳精度を実現した PCFG-LA モデルに基づく Egret[4] を用い，日本語係り受けコーパス JDC [12] (7069 文) に対して Travatar の主辞ルールで係り受け構造を句構造に変換したものを用いて学習したモデルを，ベースラインの構文解析器として使用した．構文森は 100-best 構文木に存在する hyper-edge のみで構成し，その他については枝刈りした．日本語の単語分割には KyTea [16] を用いた．単語アライメントは Nile[5] [17] を用いて行い，目的言語である英語の言語モデルは KenLM [3] を用いて 6-gram で学習した．翻訳器には Travatar [14] を用い，F2S 翻訳を行った．文単位の機械翻訳精度は BLEU+1 [7] を用いて評価した．

実験では，BLEU+1 のスコアに閾値を定め，スコアが閾値以上の文のみを自己学習に使用するよう文選択を行った．また，従来の自己学習手法である，構文解析器の出力をそのまま自己学習に使用する手法も比較した．この際，自己学習に使用する文はコーパス中からランダムに選択した文とした．

構文解析器の精度評価のために，ASPEC に含まれる日英対訳データの内，Test セット中の 100 文を人手でアノテーションを行い，正解構文木を作成した．その後，各構文解析器の精度を Egret および Evalb[6] を用いて計測した．評価には，再現率，適合率，およびそれぞれの調和平均である F 値を用いる．

5.2 実験結果

表 1 に構文解析器の精度評価結果を示す．表中の「文数」は自己学習に使用した文数を示し，JDC の文数は含まない．また，各手法の F 値について，ブートストラップ・リサンプリング法 [6] を用いて統計的有意差を検証した．

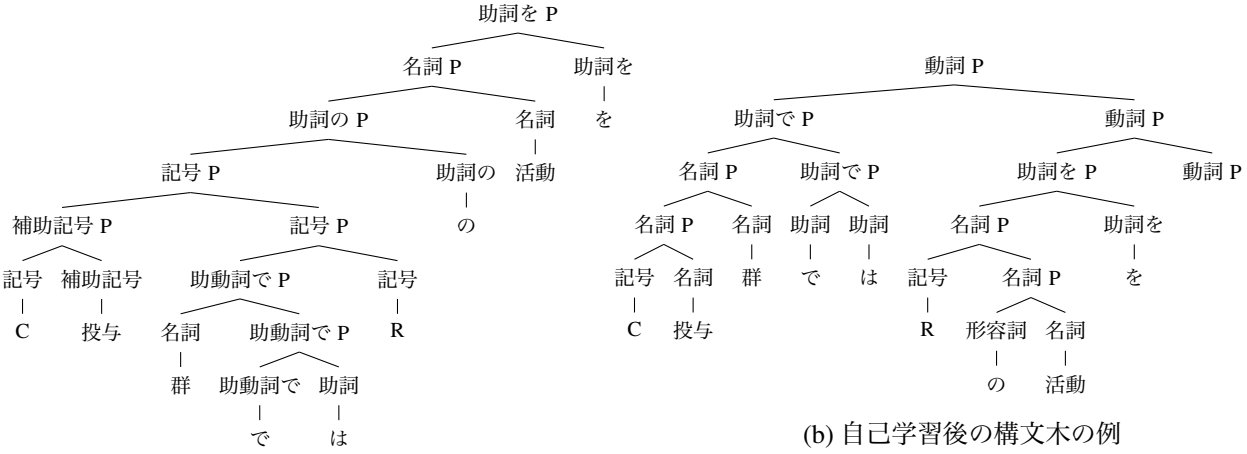
実験により，既存の自己学習手法はベースラインと比較して，95%水準で有意に精度が向上していることがわかった．これに加えて，提案手法により標的自己学習を行った場合は，99%水準で有意に精度が向上している．また，既存の自己学習手法との有意差を検証したところ，(c), (d) の手法では，95%水準で有意に精度が向上しており，提案手法により従来手法以上の精度向上が達成できていることがわかった．

構文解析器の自己学習により，改善された構文木の例を図 3 に示す．ここでは，「C 投与群では R の活動を 240 分にわたって明らかに増強した。」という文の一部を示している．この文には「C 投与群」,「R の活動」という 2 つの名詞句が含まれているが，ベースラインの構文解析器ではこれを正しく解析できていない．

[3] http://lotus.kuee.kyoto-u.ac.jp/ASPEC/
[4] http://code.google.com/p/egret-parser/
[5] http://jasonriesa.github.io/nile/
[6] http://nlp.cs.nyu.edu/evalb

表 1: 自己学習した日本語構文解析器の精度

	自己学習手法	文数	再現率	適合率	F 値	(a) との有意差	(b) との有意差
(a)	ベースライン (JDC のみ)	—	84.88	84.77	84.83	—	—
(b)	既存自己学習手法	96k	86.52	86.41	86.46	あり (95%水準)	—
(c)	BLEU+1 $\geq$ 0.7	206k	88.13	88.01	88.07	あり (99%水準)	あり (95%水準)
(d)	BLEU+1 $\geq$ 0.8	120k	88.13	88.01	88.07	あり (99%水準)	あり (95%水準)
(e)	BLEU+1 $\geq$ 0.9	58k	87.29	87.13	87.23	あり (99%水準)	なし



(a) 自己学習前の構文木の例

(b) 自己学習後の構文木の例

図 3: 自己学習による構文木の改善例

一方，自己学習後の構文解析器ではこれを正しく解析できており，精度向上につながっている．構文解析結果が改善された例を確認したところ，このように名詞句が正しく解析できた例が多く見受けられた．これは McClosky ら [10] が報告していたように，既存モデルである JDC で既知の単語が，ASPEC で異なる文脈で現れた際に解析精度が向上した結果であると思われる．

6 おわりに

本稿では，対訳コーパスを用いた構文解析器の標的自己学習を行い，解析精度の変化を検証した．実験の結果，提案手法により既存の自己学習手法を有意に上回る精度向上を達成することができた．構文解析結果が改善された例を調査したところ，特に名詞句において改善が見られた．これは，既存モデルで既知の単語が自己学習用のデータで異なる文脈で現れた際に解析精度が向上した結果であると思われる．今後は，さらに幅広い分野に対して提案手法による標的自己学習が適用可能であるかを検証したいと考えている．また，新たな構文木および文の選択手法についても検討したい．

謝辞

本研究の一部は，JSPS 科研費 25730136 の助成を受け実施したものである．

参考文献

[1] Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proc. AAAI*, pp. 598–603, 1997.
[2] Daniel Gildea. Corpus variation and parser performance. In *Proc. EMNLP*, pp. 167–202, 2001.
[3] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proc. WMT*, pp. 187–197, 2011.
[4] Zhongqiang Huang and Mary Harper. Self-training PCFG grammars with latent annotations across languages. In *Proc. EMNLP*, pp. 832–841, 2009.
[5] Jason Katz-Brown, Slav Petrov, Ryan McDonald, Franz Och, David Talbot, Hiroshi Ichikawa, Masakazu Seno, and Hideto Kazawa. Training a parser for machine translation reordering. In *Proc. EMNLP*, pp. 183–192, 2011.
[6] Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP*, pp. 388–395, 2004.
[7] Chin-Yew Lin and Franz Josef Och. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proc. COLING*, pp. 501–507, 2004.
[8] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn treebank. *Computational linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.
[9] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proc. HLT*, pp. 152–159, 2006.
[10] David McClosky, Eugene Charniak, and Mark Johnson. When is self-training effective for parsing? In *Proc. COLING*, pp. 561–568, 2008.
[11] Haitao Mi and Liang Huang. Forest-based translation rule extraction. In *Proc. EMNLP*, pp. 206–214, 2008.
[12] Shinsuke Mori, Hideki Ogura, and Tetsuro Sasada. A Japanese word dependency corpus. In *Proc. LREC*, pp. 753–758, 2014.
[13] Makoto Morishita, Koichi Akabe, Yuto Hatakoshi, Graham Neubig, Koichiro Yoshino, and Satoshi Nakamura. Parser self-training for syntax-based machine translation. In *Proc. IWSLT*, pp. 232–239, 2015.
[14] Graham Neubig. Travatar: A forest-to-string machine translation engine based on tree transducers. In *Proc. ACL Demo Track*, pp. 91–96, 2013.
[15] Graham Neubig and Kevin Duh. On the elements of an accurate tree-to-string machine translation system. In *Proc. ACL*, pp. 143–149, 2014.
[16] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. ACL*, pp. 529–533, 2011.
[17] Jason Riesa and Daniel Marcu. Hierarchical search for word alignment. In *Proc. ACL*, pp. 157–166, 2010.
[18] Hui Zhang and David Chiang. An exploration of forest-to-string translation: Does translation help or hurt parsing? In *Proc. ACL*, pp. 317–321, 2012.