統計翻訳における統語的ラベル細分化の検討

須藤 克仁　進藤 裕之　塚田 元　永田 昌明
NTTコミュニケーション科学基礎研究所

sudoh.katsuhito@lab.ntt.co.jp

概要

本稿では，統語情報に基づく統計翻訳(syntax-based statistical machine translation)の改善のために，当該手法で使用する統語的ラベルの細分化について検討する．統語的ラベルの細分化は構文解析の際に句の異なる統語的役割を区別するために有用であり，統語情報に基づく機械翻訳においても有効であることが期待される．本稿では，統語的情報を付加することで統語的ラベルを細分化する方法について述べるとともに，日英翻訳実験における比較結果について報告する．

1　はじめに

昨今の統計的機械翻訳(Statistical Machine Translation: 以後，統計翻訳)技術の中で，統語情報に基づく統計翻訳(syntax-based SMT) [1, 2] は 統語的に妥当な翻訳を行うための有望な手法と言える．

統語情報に基づく翻訳においては，「木構造」と「統語的カテゴリ」を制約として利用できる．木構造を利用することで，従来の句に基づく翻訳 [3] では扱えなかった 言語の階層構造を考慮することができ，長距離の並べ替えを自然に解くことができる．構造的制約はITG (Inversion Transduction Grammar) [4] や 階層的な句に基づく統計翻訳 [5] の主要な特徴であり，対訳文における単語対応から得られる同期文脈自由文法を利用している．この手法は言語によらず適用可能であるという利点があるものの，名詞句，動詞句といった統語的カテゴリに基づくものではないため，統語的制約という面では不十分である．統語情報に基づく翻訳では，構造的制約と統語的カテゴリの両方を制約として，構造や終端記号（単語）が同じでも異なる統語的役割を持つ部分木を明に区別することができ，翻訳の統語的妥当性の面で有利と言える．本稿はこの統語的カテゴリに基づく制約に着目する．

英語の構文解析で用いられる統語的カテゴリはPenn Treebankで利用されているラベルセットを用いることが一般的であるが，主語として用いられる名詞句も目的語として用いられる名詞句も同じ"NP" であり，分類としては比較的粗いと考えられる．Klein and Manning [6]は このような異なる統語的役割を持つ句を区別した正確な構文解析を行うために，言語学的知見に基づく統語的ラベルの細分化を行った．Matsuzaki et al. [7] や Petrov et al. [8] は 統語的ラベルを自動的に細分化する手法を提案した．このようなラベル細分化は中国語の単語列から英語の構文木への(string-to-tree: S2T)翻訳においても 有効であると報告されている [9]．また，それとは異なるアプローチとして，主辞駆動型句構造文法(Head-driven Phrase Structure Grammar: HPSG)に基づく構文解析手法が提案されている [10]．この手法は品詞ラベルや句ラベルで表現される統語的カテゴリに加えて，統語的役割や句同士の関係をより詳細に記述する言語素性を考慮することで構文解析の精度を向上させている．また，こうした言語素性が構文木から単語列への(tree-to-string: T2S)翻訳において 有効であることが示されている [11]．

本稿では，後者のアプローチで利用されているような詳細な言語素性に基づいて統語的ラベルを細分化し，S2T翻訳において利用する手法について検討する．統語的ラベルの細分化はS2T翻訳でも有用であることが期待される一方，過剰な細分化によってデータスパースネスの問題も起こり得る．そうした観点から，本稿では統語的ラベルの細分化の複数の方法を比較し，どのような細分化が有効であるかを調べる．

2　統語的ラベルの細分化方法

S2T翻訳では，翻訳規則として原言語側の句と目的言語側の部分木のペアを利用する．部分木では非終端記号として品詞ラベルや句ラベルが用いられ，当該非終端記号部分に生成される単語や句の統語的カテゴリを制約している．

本稿ではS2T翻訳規則の学習に，下記の構文解析器によって得られる構文木を用い，解析器から得られる情報を用いて異なる統語的ラベルの細分化を行った結果を比較する．

- HPSGに基づく構文解析器 Enju[1]
- 確率的CFGに基づく構文解析器 Berkeley parser[2]

[1]http://www.nactem.ac.uk/tsujii/enju/index.ja.html
[2]http://code.google.com/p/berkeleyparser/

```
<sentence id="s0" parse_status="success" fom="9.71747">
 <cons id="c0" cat="S" xcat="" head="c2" schema="subj_head">
  <cons id="c1" cat="NP" xcat="" head="t0">
   <tok id="t0" cat="N" pos="PRP">He</tok>
  </cons>
  <cons id="c2" cat="VP" xcat="" head="c3" schema="head_comp">
   <cons id="c3" cat="VX" xcat="" head="t1">
    <tok id="t1" cat="V" pos="VBD">lost</tok>
   </cons>
   <cons id="c4" cat="NP" xcat="" head="c6" schema="spec_head">
    <cons id="c5" cat="DP" xcat="" head="t2">
     <tok id="t2" cat="D" pos="PRP$">his</tok>
    </cons>
   <cons id="c6" cat="NX" xcat="COOD" head="c7" schema="coord_left">
     <cons id="c7" cat="NX" xcat="" head="t3" sem_head="t3">
      <tok id="t3" cat="N" pos="NN">wallet</tok>
     </cons>
    <cons id="c8" cat="COOD" xcat="" head="c9" schema="coord_right">
       <cons id="c9" cat="CONJP" xcat="" head="t4" sem_head="t4">
        <tok id="t4" cat="CONJ" pos="CC">and</tok>
       </cons>
       <cons id="c10" cat="NX" xcat="" head="t5" sem_head="t5">
        <tok id="t5" cat="N" pos="NN">umbrella</tok>
       </cons>
      </cons> </cons> </cons> </cons> </cons>
.
</sentence>
```

図 1: Enjuによる英語構文解析結果の例(XML形式, 一部の属性は省略).

2.1 Enjuと詳細な素性に基づく細分化

Enjuは図1に示すような構文木を解析結果として出力する. 図中, cons は構成素(constituent), tok は単語, cat は統語的カテゴリ[3]を示す. 本稿では属性 cat に利用されているラベル(22種類)を基本ラベルとして利用し, その他3つの属性 xcat (統語的カテゴリの追加属性: 属性値なしを含む9種類), head (統語的主辞), schema (スキーマ: 12種類) を基本ラベルの細分化のために利用する. ラベルの細分化は単純に追加属性のラベルを連結する形で行う. なお, head については主辞が左側の子である(left)か, 右側の子である(right)であるかの二値で表現する. 例えば基本ラベルをスキーマの情報を利用して細分化するのであれば, "VP+head_comp" のようなラベルを利用する. xcat や headに相当する情報は従来の構文解析や統語情報に基づく翻訳でも利用されてきたが, schemaは兄弟となる構成素の関係を記述する言語素性であり, S2T翻訳において句の統語的役割をより詳細に同定できるという面で有用であることが期待できる. 後述する実験では, 上記3つの追加属性のすべての組み合わせについて比較実験を行う.

2.2 Berkeley parserを用いた細分化

Berkeley parserは図2に示すような構文木を解析結果として出力することができる[4]. ここで, 句および品詞のラベルには数字が付加され細分化されている(例: NP-32). 本稿では, 細分化されていない句や品詞のラベル(例: NP)を基本ラベルとして扱う. 本稿で用いた英語解析の文法は細分化レベル6 (Berkekey parserと共に配布されている英語解析モデル)である. Berkeley parserでは上記Enjuのような詳細な言語素性は得られないが, 言語学的知見に基づく細分化[6, 9]や自動細分化[7, 8]に基づく比較実験を行った. まず, 言語学的知見に基づく細分化については, Wang et al.[9]に基づいて, 前置詞を細分化する SPLIT-IN と 動詞句を細分化する SPLIT-VP について比較した. また, 自動細分化については, Berkeley parserの自動細分化ラベルをそのまま利用した.

```
( (S-0
    (@S-26
      (NP-32 (PRP-4 He) )
      (VP-9
        (VBD-5 lost)
        (NP-46
          (@NP-5
            (NP-4 (PRP$-2 his) (NN-33 wallet) )
            (CC-6 and) )
          (NP-45 (NN-10 umbrella) ) ) ) )
    (.-0 .) ) )
```

図 2: Berkeley parserによる英語構文解析結果の例.

3 実験

実験により, 異なる細分化レベルの統語的ラベルがS2T翻訳にどのような効果をもたらすかを調べた. 本実験ではS2T翻訳の実装として, 統計翻訳ツールキットMoses[5] に含まれる moses_chart による統語拡張型翻訳 (syntax-augmented SMT) [13] を利用した. moses_chart は階層的句に基づく翻訳 [5] や統語拡張型翻訳に対応する翻訳デコーダである [6].

3.1 実験条件

実験に用いたデータはNTCIR-9 PatentMT [15]の日英特許翻訳データセット (学習データ 約320万文対, 開発データ 2,000文対, テストデータ 2,000文対)である. 単語分割は日本語側はMeCab (ipadic)[7]を, 英語側はEnjuによる単語分割結果を利用し, Berkeley parserを用いる際にはEnjuの単語分割結果を入力として構文解析を行った. S2T

[3]詳細は Enjuのラベル定義[12] を参照.
[4]オプション "-binarize -viterbi -substates" を利用する.
[5]http://www.statmt.org/moses/
[6]通常S2T翻訳は翻訳規則の目的言語側に任意の深さの部分木を利用可能なものを指すが[14], 統語拡張型翻訳は階層的句に基づく翻訳における非終端記号に統語的ラベルを利用する拡張であり, 翻訳規則の目的言語側は常に深さ1である
[7]http://mecab.sourceforge.net

翻訳モデルの学習はMosesを用いた標準的な手順(MGIZA++とgrow-diag-finalヒューリスティクスによる単語対応付け，S2T翻訳規則の抽出(構文木のスパンは無制限とした)，S2T翻訳確率のGood-Turingディスカウント)を行い，言語モデルには英語の単語5-gramを利用した．本実験で比較する手法の違いは，S2T翻訳モデル学習で利用される英語側構文木及びその統語的ラベルの違いによるものである．

3.2 評価尺度

翻訳精度の評価には BLEU [16][8] 及び RIBES [17][9] を利用した．BLEUでは単語n-gramを用いた局所的な翻訳精度が重視されるのに対し，RIBESでは大域的な語順の正しさが重視されるという違いがある．

また，S2T翻訳モデル自体の良し悪しを推定するために，テストセットの翻訳時に適用可能なS2T翻訳規則[10]の数，及びS2T翻訳規則の日本語側に対する英語側の条件付きエントロピーを用いた．条件付きエントロピー $H(E|J)$ は以下のように計算される．

$$H(E|J) = -\frac{1}{\|P_j\|} \sum_{j \in P_j} \sum_{e \in P_e(j)} P(e|j) \log P(e|j) \quad (1)$$

ここで，P_jはS2T翻訳規則集合に含まれる日本語の階層的句の集合，$P_e(j)$は日本語の階層的句jから翻訳可能な英語の階層的句の集合を表す．この条件付きエントロピーはS2T翻訳における翻訳の曖昧性を表す．エントロピーが大きければ，多くの翻訳候補が拮抗しており翻訳が難しくなることを意味する．

3.3 実験結果

図1にS2T翻訳規則の統計と翻訳精度を示す．図上方のEnjuに基づく方法では，統語的ラベルをheadやschemaといった属性を用いて細分化することで翻訳精度が向上している．逆に統語的ラベルを利用せず，木構造のみを利用(統語的ラベルを Xに統一)した場合，翻訳精度が大きく低下している．これらの結果から，S2T翻訳規則における統語的ラベルは翻訳デコーダがより翻訳候補を選択するために重要な役割を果たしており，木構造の制約のみではS2T翻訳は十分な性能を発揮できないことが分かる．また，本実験で用いた統語的ラベルの細分化のための属性のうち，特にschemaが有効であり，逆にxcatはBLEU,RIBESとも向上の度合いが非常に小さい．統語的ラベルの細分化によって適用可能なS2T翻訳規則の数は非常に大きくなっているが，一方で規則のエントロピーは低下，翻訳精度は向上する傾向にある．一方，Berkeley parserの解析に基づくS2T翻訳では，Enjuを用いた場合と比較して一定した効果は得られなかった．統語的ラベルを利用することの効果はEnjuの場合と同様であるものの，細分化で目立った性能向上は得られなかった．SPLIT-INやSPLIT-VPではS2T翻訳規則数が増えたもののエントロピーが増大し，翻訳精度も低下している．一方の自動細分化では規則数が大きく増加したことで規則自体の曖昧性は平均的には低下しているが，BLEUが低下，RIBESが向上という結果になった．

3.4 議論

EnjuとBerkeley parserの結果の違いの一つの理由として，統語的ラベルの細分化にどのような情報が利用されているかが異なることが考えられる．Enjuのheadやschemaの属性は「左右どちらの子が統語的主辞であるか」「左右の子はどのような統語的関係にあるか(主語と述語，述語と保護，など)」を示すことから，句の統語的機能・役割の細分化に有効であることが期待できる．本稿の実験結果は英日翻訳における事前並べ替えにおいて統語的主辞の利用が非常に有効であった結果とも符合する．

一方，Berkeley parserで用いた細分化はそこまでの分類に寄与していないことが予想される．まず，SPLIT-INやSPLIT-VPによる細分化によって区別される語句の影響は限定的と言え，また，エントロピーが若干増大している点からも本実験におけるS2T翻訳の改善には不十分であったと思われる．また，自動細分化によって翻訳精度が改善できなかった理由としては，統語的意味を持たない細分化であること，また細分化の度合いが大きいことから過学習しやすいことが考えられる．後者の原因については，エントロピーが低下していることから分かる通りある非終端記号からの導出の曖昧性は平均的には低下しているものの，翻訳規則数がほぼ2倍となっていることから過学習と同様の問題があると考えることができる．以上から，S2T翻訳においてはスキーマで表現される子の統語的関係は有効な素性であり，そういった素性を獲得・活用することが統計翻訳にも貢献することが分かった．なお，本稿の実験ではBerkeley parserの細分化ラベルをそのまま利用したが，Enjuにおいて有効であった主辞などの情報を利用することで精度を改善できる可能性もあり，今後の検討課題としたい．

4 おわりに

本稿では，S2T翻訳において統語的ラベルの細分化について複数の手法を比較・検討した．Enjuを用い

[8] BLEUの計算はmulti-bleu.perlを利用した．
[9] http://www.kecl.ntt.co.jp/icl/lirg/ribes
[10] テストセットに適用可能な翻訳規則の抽出はfilter-rule-table.pyを用いた．

表 1: S2T翻訳規則に関する統計と, BLEUとRIBESによる翻訳評価結果 (**太字**は最善の結果を示す).

構文解析器	統語的ラベル	条件付きエントロピー	規則数	BLEU (%)	RIBES (%)
Enju	None (Xに統一)	0.815	2.87M	30.6	72.7
	標準ラベル	0.736	4.84M	31.1	73.3
	+head	0.487	5.41M	31.4	73.8
	+schema	0.470	5.55M	**31.8**	73.8
	+xcat	0.513	5.07M	31.3	73.4
	+head+schema	0.467	5.55M	31.7	73.9
	+head+xcat	0.479	5.48M	31.5	73.6
	+schema+xcat	0.468	5.56M	31.6	73.9
	+head+schema+xcat	0.466	5.57M	**31.8**	**74.2**
Berkeley	None (Xに統一)	0.687	2.30M	28.8	71.5
	標準ラベル	0.412	3.32M	29.4	72.1
	SPLIT-IN	0.435	3.42M	29.3	71.6
	SPLIT-VP + SPLIT-IN	0.447	3.43M	29.0	71.5
	自動細分化 (-substates)	0.355	6.47M	28.5	72.5

た実験では，詳細な統語的属性(schema)を標準のラベルに付加して細分化を行うことで， S2T翻訳精度を改善できることを示した． また，Berkeley parserを用いた実験では良好な結果は得られなかった． これらの結果は，S2T翻訳における統語的ラベルの細分化において， 句同士の統語的な関係を利用することが有効であることを示している． 今後の課題としては，本稿の知見に基づく統語的機能・関係に着目した自動細分化手法の改善や， このようなアプローチの他言語への適用などが考えられる．

参考文献

[1] K. Yamada and K. Knight. A Syntax-based Statistical Translation Model. In Proc. ACL, pp. 523--530, 2001.

[2] M. Galley, M. Hopkins, K. Knight, and D. Marcu. What's in a translation rule? Proc. HLT-NAACL, pp. 273--280, 2004.

[3] P. Koehn, F. J. Och, and D. Marcu. Statistical Phrase-based Translation. In Proc. HLT-NAACL, pp. 263--270, 2003.

[4] D. Wu. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. Computational Linguistics, 23(3):377--403, 1997.

[5] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In Proc. ACL, pp. 263--270, 2005.

[6] D. Klein, C. D. Manning. Accurate Unlexicalized Parsing. In Proc. ACL, pp. 423--430, 2003.

[7] T. Matsuzaki, Y. Miyao, and J. Tsujii. Probabilistic CFG with Latent Annotations. In Proc. ACL, pp. 75--82, 2005.

[8] S. Petrov, L. Barrett, R. Thibaux, and D. Klein. Learning Accurate, Compact, and Interpretable Tree Annotation. In Proc. COLING-ACL, pp. 433--440, 2006.

[9] W. Wang, J. May, K. Knight, and D. Marcu. Re-structuring, Re-labeling, and Re-aligning for Syntax-Based Machine Translation. Computational Linguistics, 36(2):247--277, 2010.

[10] Y. Miyao and J. Tsujii. Feature Forest Models for Probabilistic HPSG Parsing. Computational Linguistics, 34(1):35--80, 2008.

[11] X. Wu, T. Matsuzaki, and J. Tsujii. Fine-Grained Tree-to-String Translation Rule Extraction. In Proc. ACL, pp. 325--334, 2010.

[12] Enju Output Specifications. http://www.nactem.ac.uk/tsujii/enju/enju-manual/enju-output-spec.html.

[13] A. Zollmann and A. Venugopal. Syntax Augmented Machine Translation via Chart Parsing. In Proc. WMT, pp. 138--141, 2006.

[14] H. Zhang, L. Fang, P. Xu, and X. Wu. Binarized Forest to String Translation. In Proc. ACL-HLT, pp. 835--845,

[15] I. Goto, B. Lu, K. P. Chow, E. Sumita, and B. K. Tsou. Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop. In Proc. of NTCIR-9, pp. 559--578, 2011.

[16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proc. ACL, pp. 311--318, 2002.

[17] 平尾, 磯崎, Duh, 須藤, 塚田, 永田. RIBES: 順位相関に基づく翻訳の自動評価法. In 言語処理学会 第17回年次大会発表論文集, pp. 1115--1118, 2011.