統計翻訳における構文木を用いた語順制約の導入 *

山本博史, 大熊英男，隅田英一郎 (NICT/ATR)

1　はじめに

近年広く使われるようになってきている翻訳手法である統計翻訳では，一般的にフレーズをベースとしたモデル (PBSMT)[1][2][3] が用いられる．PBSMT における問題の一つが語順並び替えであり，その中でも長距離の語順並び替えは特に深刻な問題である．この語順並び替えの問題に対しては様々な手法が提案されているが，大きく二つのタイプに分けることができる．一つ目は構文情報は直接翻訳モデルの中に組み込むもので，語順並び替えに対する詳細な情報をモデル化できると考えられるものの，モデル構築には大量の訓練データを必要とする問題点がある．二つ目は語順制約を翻訳モデルとは別に与えるもので，IBM 制約 [4]， Lexical word reordering model[5]， Inversion transduction grammar (ITG) 制約 [6][7] がこのタイプに属する．これらの手法の制約力は構文構造を直接制約に取り込むことができないため，一つ目のタイプに比べると弱いと考えられるが，大量の訓練データを必要とするという問題点は生じない．

本稿では ITG 制約の拡張として翻訳元文の構文木情報を直接語順並べ替えの制約に組み込む Imposing source tree on ITG (IST-ITG) 制約を提案する．ITG 制約では翻訳元文の木構造として特定のものを仮定していないのに対し，翻訳元文を構文解析して得られる木構造を用いるため，より強い制約を与えることができる．たとえば，4 単語からなる翻訳元文 $\{abcd\}$ に対しては可能なすべての並び替えは 4! 通りであるが， ITG 制約は $\{CADB\}$ と $\{BDAC\}$(以降，各大文字は小文字単語の対訳を表すものとする) の二つの並びしか排除できないため，$22(4!-2)$ 通りの語順を考えなければならない．これに対し，IST-ITG 制約では， $8(2^3)$ 通りの語順に減らすことができる．

本稿では 2 章ではまず，単語ベースモデルにおける IST-ITG について述べ，3 章でフレーズベースモデルへの拡張を行う．4 章で提案法を用いた実験結果について述べ，5 章でまとめを行う．

2　IST-ITG 制約

まず，語順並び替え制約に関する先行研究として 3 手法を単語モデルで，かつ，翻訳元と翻訳先の対訳単語が一対一対応する場合の説明を行う．

2.1　IBM 制約

この制約では，次式に示されるように，並び替えの距離に応じたペナルティがかけられる．

$$p_D = exp(-\sum_i d_i) \tag{1}$$

ここで， d_i は各のように定義される．

$$d_i = abs(position(e_{i-1}) + 1 - position(e_i)) \tag{2}$$

ここで e_i は，翻訳元文で先頭から i 番目の単語 f_i から翻訳された翻訳先言語の単語を表す． $position(w)$ は単語 w の文頭からの位置を表す．仏英翻訳のように類似言語対の翻訳では d_i に対してしばしば制限がかけられるが，日英や中英といった言語対ではこの制限はかけない方が良い場合が多い．

2.2　Lexical Reordering Model

Lexical reordering model，では対訳ペア $\{f_i, e_i\}$ ごとに語順並べ替えに対する確率が割り当てられる．語順並べ替えは monotone，swap，discontinuous の 3 通りに分類され，それぞれの並べ替えが起こる確率が付与される．monotone は並び替えが起こらない場合 (a b が A B 翻訳)，swap は語順が逆転する場合 (a b が B A に翻訳)，discontinuous はそれ以外の並び替えである．これら 3 つの確率は前接する語，後続する語それぞれに対して与えられるため，一対訳ペアにつき 6 つの確率が割り当てられることになる．前接する場合に対する確率を p_r，後続する場合を p_l，monotone を m，swap を s，discontinuous を d と表記すると，翻訳元単語並び f_{i-1}, f_i から生成される翻訳先単語 e_{i-1}, e_i に位置関係は次式で表される．

- $p(e_{i-1}, e_i) = p_r(m|f_{i-1}, e_{i-1})p_l(m|f_i, e_i)$
- $p(e_i, e_{i-1}) = p_r(s|f_{i-1}, e_{i-1})p_l(s|f_i, e_i)$
- $p(otherwise) = p_r(d|f_{i-1}, e_{i-1})p_l(d|f_i, e_i)$

2.3　ITG 制約

一対一対応の単語モデルの場合，並び替えに対して制約がなければ N 単語からなる翻訳元文に対し，$N!$ 通りの語順を翻訳先言語で考えなければならない．ITG 制約では次のような制約をかけることで，考慮すべき語順を減らすことができる．

* Introduction of Word Order Constraints Using Syntax Tree for SMT. by YAMAMOTO, Hirofumi OKUMA Hideo SUMITA, Eiichiro (NiCT/ATR)

- 翻訳元文に対し，可能なすべてのバイナリ木を考える．
- 翻訳先文の語順はバイナリ木の任意のノードを回転させることで得られる．

$N = 4$ の場合，ITG 制約は並び替えの組み合わせを $4! = 24$ 通りから 22 通りに減らすことができる．この差は N が大きくなる程大きくなり， $N = 10$ では減少率は $206,098/3,628,800 = 0.0568$ となる．

2.4 Imposing Source Tree

ITG 制約では，翻訳元文に対しバイナリ木を特定していない．従って，翻訳元文を構文解析することによってバイナリ木を特定できればより強い制約を与えることができると考えられる．提案手法では構文木の内，bracketed sentence の情報のみを用いることにする．たとえば英日翻訳における翻訳元文「This is a pen.」に対しては ((This) ((is) ((a) (pen))) (.)) という構造を利用する．この構造はバイナリ木と等価である．この制約を「imposing source tree on ITG」(IST-ITG) とよぶことにする．IST-ITG を用いた場合，$N = 4$ の場合，ITG の 22 通りの語順に対し 8 通りの語順を翻訳先言語で考えればよいことになる．たとえば，翻訳元文木 $((ab)(cd))$ に対しては $\{ABCD\}$，$\{BACD\}$，$\{ABDC\}$，$\{BADC\}$，$\{CDAB\}$，$\{CDBA\}$，$\{DCAB\}$，$\{DCBA\}$ の 8 通りとなる．同じく $(((AB)C)D)$ に対しては，$\{ABCD\}$，$\{BACD\}$，$\{CABD\}$，$\{CBAD\}$，$\{DABC\}$，$\{DBAC\}$，$\{DCAB\}$，$\{DCBA\}$ の 8 通りとなる．一般的には IST-ITG の下で考慮すべき語順は 2^{N-1} となる．

2.5 非バイナリ木への拡張

前節では，翻訳元文の構造としてバイナリ木を仮定した．しかしながら，構文解析の結果が常にバイナリ木になるとは限らない．このような場合，構文木のノードには 3 つ以上の枝を持つものが存在することになる．このような場合，同一ノードを持つ枝の間での語順の入れ替えは自由であるとするが，ITG 制約は適用するものとする．たとえば，非バイナリ木 $(a(bcd))$ に対しては，(bcd) 内の入れ替えで 6 通り，a と (bcd) の入れ替えで 2 通り，全体で $6 \times 2 = 12$ 通りの語順が許される．また，4 つ以上の枝を持つノードに対しては ITG 制約が適用されるため，非バイナリ木 $(a(bcde))$ に対しては $22 \times 2 = 44$ 通りの語順が許される．一般的な語順の数は次式で表される．

$$\prod_{i=1}^{n}(S_{Bi}) \tag{3}$$

ここで，S_k は $N = k$ のときの ITG 制約のもとでの可能な語順の数を表し，$N = 3$ では 6 通り，$N = 4$ では 22 通りである．また，Bi は i 番目のノードの枝の数である．

3 IST-ITG のフレーズベースモデルへの拡張

前章では単語モデルにおける IST-ITG について述べた．本章ではこれをフレーズベースモデルに対して拡張する．ITG をフレーズベースモデルへの拡張に拡張する場合は単に単語のバイナリ木をフレーズのバイナリ木で置き換えればよい．しかしながら，IST-ITG では構文解析結果に木を用いるため，木のノードは単語でなければならない．一方，フレーズベースモデルではフレーズを単位として翻訳が行われるため，制約の単位との不一致が生じる．この不一致を埋めるために，次の制限を導入する．

- フレーズが二つ以上に分割されるような語順入れ替えは許されない．

この制限を新たに導入することによって語順入れ替えは必ずフレーズ単位で行われるようになるため，単語モデルの場合と同じように語順入れ替えに制約をかけることができるようになる．たとえば，木 $((abc)((de)(fg)))$ に対し，b，c，d がフレーズ ph をなしている時 (図 1 参照)，ノード 1 に対しては語順入れ替えを行うことができない．理由はこのノードはフレーズ ph の一部 b，c を含んでいるため，入れ替えを行うと ph が二つに分割されてしまうためである．同様の理由でノード 2 と 4 に対しても入れ替えを行うことはできない．またノード 3 はフレーズの一部を含んでいないため，ノード 5 はフレーズ全体を含んでいるため語順入れ替えを行うことができる．結果として，この木から許される翻訳先文の語順は $\{ A\ PH\ E\ F\ G \}$，$\{ A\ PH\ E\ G\ F \}$，$\{ G\ F\ E\ PH\ A \}$，$\{ F\ G\ E\ PH\ A \}$ の 4 通りである．

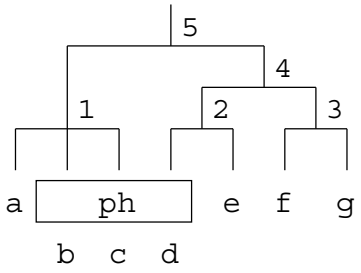

Fig. 1 Example sentence tree with phrase

4 評価実験

4.1 評価コーパス

最初に英日特許コーパスを用いた評価実験を行った．コーパスの詳細を表 1 に示す．このコーパスは対訳文の自動アライメント [8] を用いて作成されたもので，アライメントスコアの高いほうから 900 文を評価セットに (参照訳の数は 1 となる)，次の 1,000 文をデベロップメントセットに，残りを訓練セットとして用いた．このコーパスは NTCIR-7 ワークショップ [9] の特許翻訳タスクにおける訓練コーパスのサブセットとなっている．

Table 1　日英特許コーパス

	# of sent.	Total words	# of entries
E/J Train	10M	273M/257M	797K/282K
E/J Dev	1,000	39K/37K	4,971/4,614
E/J Eval	900	29K/32K	3,967/3,683

4.1.1 英日翻訳実験

まず，英日方向の翻訳実験を行った．フレーズベース翻訳モデルの学習には GIZA++ ツールキット [10] を，言語モデルの学習には SRI language model ツールキット [11] を用いた．言語モデルは 5gram であり，Kneser-Ney 平滑化 [12] を用いている．翻訳デコーディングのためのパラメータチューニングには 1,000 文のデベロップメントセットを用いて minimum error training[13] 2003) を行った．英語の構文解析は Charniak パーザ [14] を，日本語のセグメンテーションには 茶筅 [15] を用いた．デコーダは我々が独自に開発した Pharaoh[17] 上位互換デコーダである CleopATRa を用いた．CleopATRa には IST-ITG 制約下でデコーディングを行うアルゴリズム [16] が実装されている．

実験に用いた語順入れ替えに対する制約の組み合わせは次に示す通りである．「Monotone」:語順入れ替えなし，「No constrains」:語順入れ替え制約なし，「IBM」:IBM 制約，「ITG」:ITG 制約，「IBM+ITG」:IBM と ITG の併用，「IBM+LR」: IBM と Lexical reordering model の併用,「IST」:提案法である IST-ITC の単独使用，「IBM+IST」:IBM と IST-ITC の併用,「IBM+LR+IST」:IBM，Lexical reordering model，IST-ITG の併用．

表 2 に評価結果を示す．ITG 制約 (ITG) との比較において，提案法 IST-ITG (IST) は BLEU で 2.67，WER で 5.39%性能が向上している (表中太字)． WER における性能向上が特に大きく，本制約がグローバルな語順入れ替えに対する制約として，うまく働いていることがわかる．また，ITG および IST-ITG に対し IBM を併用した場合 (IBM+ITG と IBM+IST)，BLEU で 1.57，WER で 4.63%の性能向上，lexical reordering model と IBM との併用 (IBM+LR と IBM+LR+IST) では，BLEU で 1.03，WER で 5.12%の向上であり，いずれの場合でも特に WER において大きな性能向上を示している．

Table 2　英日特許翻訳における性能評価

	BLEU	NIST	WER	PER
Monotone	24.91	6.95	79.97	42.02
No constraint	26.83	7.19	81.10	39.52
IBM	28.35	7.29	78.35	39.25
ITG	**27.59**	**7.26**	**80.29**	**39.15**
IBM+ITG	28.50	7.30	78.01	39.29
IBM+LR	31.17	7.50	76.30	38.61
IST	**30.26**	**7.41**	**74.90**	**38.93**
IBM+IST	30.07	7.41	73.38	39.05
IBM+LR+IST	32.20	7.61	71.18	38.15

4.1.2 日英翻訳実験

続いて，同じコーパスを用いて日英方向の翻訳実験を行った．日本語の構文解析には，係り受け解析器 CaboCha[18] を用いて文節間の係り受け関係をまず抽出し，これを bracketed tree に変換した．

表 3 に評価結果を示す．ITG 制約 (ITG) との比較において，提案法 IST-ITG (IST) は BLEU で 1.21，WER で 3.81%性能が向上している．また， lexical reordering model と IBM との併用 (IBM+LR と IBM+LR+IST) では，BLEU では性能向上は見られなかったが， WER で 4.47%の向上であり，日英翻訳同様，特に WER において大きな性能向上を示している．性能向上が英日翻訳より小さかった原因としては，英語構文解析と，日本語係り受け解析の難易度の差，および英語では直接木を抽出しているのに対し，日本語では係り受け関係から変換することによって木を得ていることが考えられる．

5 まとめ

本稿ではフレーズベース統計翻訳 (PBSMT) における語順入れ替えの制約として，翻訳元文の木構造情報を利用する手法を提案した．提案手法である IST-ITG 制約は，ITG 制約の拡張であり，ITG では特定しない翻訳元文の木構造情報を取り込むことによって，より強い制約を与えることができる．たとえば，ITG で

Table 3　日英特許翻訳における性能評価

	BLEU	NIST	WER	PER
Monotone	26.29	7.25	76.42	40.85
No constraint	26.20	7.18	81.41	40.76
IBM	27.87	7.34	78.16	39.94
ITG	**27.01**	**7.24**	**80.43**	**40.50**
IBM+ITG	28.16	7.35	78.04	40.07
IBM+LR	29.93	7.54	77.27	39.12
IST	**28.32**	**7.31**	**76.62**	**40.67**
IBM+IST	28.14	7.32	74.13	40.40
IBM+LR+IST	29.77	7.50	72.80	39.73

は 4 単語からなる翻訳文に対し，22 通りの単語並び替えの組み合わせを許すが，IST-ITG ではこれを 8 通りに減らすことができる．IST-ITG は日英特許翻訳コーパスを用いた実験で，英日方向で ITG よりも BLEU で 2.7，WER で 5.7% ，日英方向では BLEU で 1.2，WER で 3.8%性能が向上している．IST-ITG は WER において特に大きな性能向上が見られ，グローバルな語順入れ替えのための制約としてうまく働いていることが確認できた．

参考文献

[1] Daniel Marcu, William Wong, "A phrase-based, joint probability model for statistical machine translation," Proc. EMNLP-2002, pp.133-139, 2002.

[2] P. Koehn, F. J. Och, D. Marcu, "Statistical phrase-base translation," Proc. HLT-NAACL, pp. 127-133, 2003.

[3] F. J. Och, H. Ney, "The alignment template approach to stattistical machine translation, Computational Linguistics, 30(4), pp417-449, 2004.

[4] A. L. Berger, P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, J. R. Gillett, A. S. Kehler, and R. L. Mercer, "Language translation apparatus and method of using context-based translation models, " United States patent, patent number 5510981, April, 1996.

[5] C. Tillmann, "A unigram orientation model for statistical machine translation," HLT-NAACL, 2004.

[6] Dekai Wu, "Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora," In Proc. IJCAI, pp. 1328-1334, Montreal, August, 1995.

[7] Dekai Wu, "Stochastic inversion transuduction grammars and bilingual parsing of parallel corpora," Computational Linguiatics, 23(3), pp.377-403, 1997.

[8] Masao Utiyama and Hitoshi Isahara, "Reliable Measures for Aligning Japanese-English News Articles and Sentences", ACL-2003, pp. 72-79, 2003.

[9] NTCIR-7
http://ntcir.nii.ac.jp/

[10] F. J. Och, H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," Computational Linguistics, No. 1, Vol. 29, pp. 19-51, 2003.

[11] A. Stolcke, "SRILM - An Extensible Language Model Toolkit," Proc. ICSLP'02, 2002.
http://www.speech.sri.com/projects/srilm/

[12] R. Kneser, H. Ney, "Improved backing-off for m-gram language model," Proceedings of the IEEE International Conference of Acoustic, Speech, and Signal processing. Vol. 1, pp. 181-184, 1995.

[13] F. J. Och, "Minimum error rate training for statistical machine trainslation," Proc. ACL, 2003.

[14] E. Charniak, "A Maximum-Entropy-Inspired Parser," Proc. NAACL-2000, pp.132-139, 2000.

[15] Chasen
http://chasen-legacy.sourceforge.jp/

[16] 山本博史, 大熊英男, 隅田英一郎,"統計翻訳における木構造制約の導入,"情報処理学会研究報告,2007-NL-181,pp. 65-70,2007

[17] P. Koehn, "PHARAOH: A beam search decoder for phrase-based statistical machine translation models," Proc. AMTA, 2004.
http://www.isi.edu/publications/licensed-sw/pharaoh/

[18] Cabocha
http://chasen.org/ taku/software/cabocha/