係り受け森を用いた統計的機械翻訳

林 克彦 †‡　渡辺 太郎 ‡　隅田 英一郎 ‡　松本 裕治 †

† 奈良先端科学技術大学院大学 情報科学研究科
‡ 情報通信研究機構

{katsuhiko-h,matsu}@is.naist.jp　{taro.watanabe,eiichiro.sumita}@nict.go.jp

1　はじめに

単語や句に基づいた統計的機械翻訳は仏英など語族の近い言語間の翻訳において大きな成果を上げてきた．しかし，中英など構文構造が異なる言語間での翻訳では単語や句の並び換えを上手く扱うことができないため，高い翻訳精度を達成することが難しかった．そこで，00 年前半から syntax directed なアプローチに基づいた統計的機械翻訳が注目され，盛んに研究され始めている [8, 5, 3, 2, 7]．

木構造を利用した統計的機械翻訳では入力と出力のどこに木構造を採るかで種類が分かれるが，入力に木構造，出力に文字列を採る仕組みが現状では最も高い翻訳精度を示している．また，構文解析の間違いや曖昧性を解消するため，複数の解析結果をコンパクトに扱うことができる packed shared forests(以下，圧縮構文森) を入力とする森に基づいた統計的機械翻訳も考案され，翻訳精度の向上が確認されている [10, 9]．

上記した木や森に基づいた方式では木構造として句構造木を採るものが一般的である．しかし，句構造では語順に自由性のある言語を解析することが難しいという問題や解析の曖昧性が生じやすいといった問題がある．一方で，係り受けと呼ばれる構文構造では単語 (または文節) 間の依存関係によってのみ文の構造が表されるため，句構造におけるような問題が生じにくいという利点がある．このため係り受けは多言語での解析が可能であり，機械翻訳のように様々な言語を扱うタスクでは句構造木よりも適切な木構造であると考えられる．

本稿では係り受けを入力，文字列を出力とする森に基づいた統計的機械翻訳を提案する．表 1 は係り受けを用いた統計的機械翻訳における本稿の位置付けを示している．表 1 で示した通り，係り受けを利用したシステムも多く提案されてきたが，それらの多くでは出力側 (あるいは両言語側) に係り受けを用いることに焦点が当てられている．また，係り受け森 (dependency forests) は Tu10 [15] で出力側に用いられただけであり，入力側に利用する研究は未だない．よって，本稿での新規性は係り受け森を入力側に用いた統計的機械翻訳システムを提案したという点にある．実験ではこの係り受け森を入力側に用いた機械翻訳システムの性能を従来のオープンソースシステムと比較することで，その有用性を示した．

以下，2 節で係り受け森，3 節で提案システムについて詳細な説明を行い，4 節で提案システムと階層句に基づいたシステムを比較した実験結果を示す．5 節では実験結果を踏まえた上でまとめを行う．

2　係り受け森

係り受けは図 1(a)，1(b) で示すように単語 (または文節) 間の依存関係によって文の構文構造を表したものである．図

表 1: 係り受けに基づいた手法における本稿の位置付け

係り受け	Tree	Forest
入力	本稿	本稿
出力	Shen08[13]	Tu10
両方	Ding05 , Quirk05 Shen09[14] , Gimpel09[4] Mi10(入力は句構造)[11]	-

1(a)，1(b) では文「i saw a girl with a telescope」に対して，それぞれ異なる係り受け解析結果を示している．二つの係り受け木のうち共通する解析部分を共有させて森として表したものを図 1(c) に示す

2.1　超グラフによる定義

係り受けの構造は，頂点をスパンが付与された単語とし，一つの単語と複数の単語を結ぶ超辺で定義した超グラフで表現できる．形式的に圧縮係り受け森 F は頂点の集合 V と超辺の集合 E を持った $\langle V, E\rangle$ となる．文を $w_{1:l} = w_1 \dots w_l$ とした場合，各頂点 $v(\in V)$ は単語 w と単語 w が支配する文内のスパン i, j から $w_{i,j}$ とできる．超辺 e は $\langle tails(e), head(e)\rangle$ で定義でき，$head(e) \in V$，$tails(e) \in V^+$ である．例えば図 1(c) から saw という単語を $head$ として持つ超辺は

$$e_1 : \langle(\mathrm{i}_{0,1}, \mathrm{girl}_{2,4}, \mathrm{with}_{4,7}), \mathrm{saw}_{0,7}\rangle$$

と表され，i，girl，with を $tail$ に持つ．

2.2　重み付き超グラフとしての表現

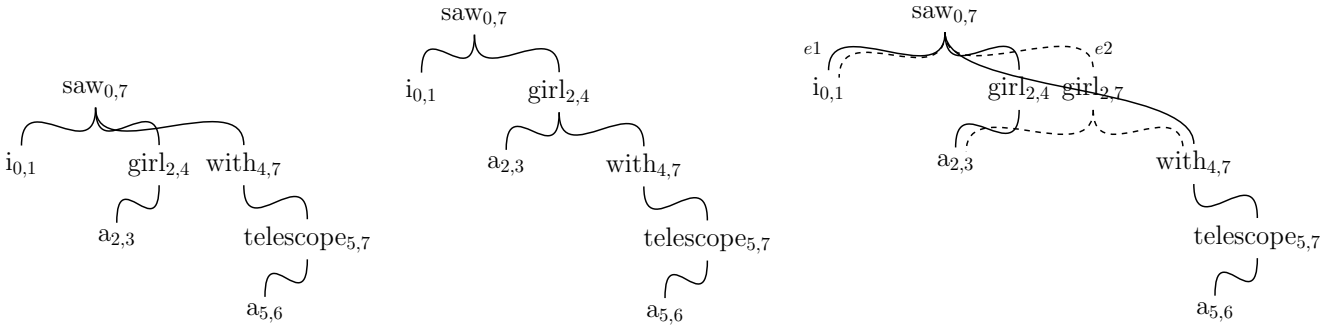
確率文脈自由文法に基づいた句構造木とは異なり，係り受けでは超グラフ上の超辺に確率を重みをとして割り当てる手法を検討する必要がある．本稿では Tu10 に従って，各超辺に確率を付与した．Tu10 の手法ではまず，各超辺の出現回数 $c(e)$ に

$$c(e) = \exp \frac{\sum_{v \in tails(e)} s(v, head(e))}{|tails(e)|} \tag{1}$$

として正の値を割り当てている．ここで $s(v1, v2)$ は $v2$ を $head$，$v1$ を $tail$ としたときの係り受け解析モデルのスコアを返す関数である．次にこの回数に基づいて確率 $p(e)$ を

$$p(e) = \frac{c(e)}{\sum_{e' : head(e') = head(e)} c(e')} \tag{2}$$

として推定する．



(a) 係り受け木 1　(b) 係り受け木 2　(c) 係り受け森

図 1: 英語文に対する係り受け木と係り受け森

3　係り受け森を用いた翻訳方式

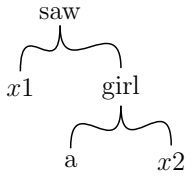
本稿で提案するシステムは形式的に $\langle \Sigma, \Delta, \mathcal{R} \rangle$ から成る．Σ は入力側のアルファベット，Δ は出力側のアルファベット，R は規則の集合である．英日翻訳を例とした場合，Σ は英語側の単語集合であり，Δ は日本語側の単語集合をとる．

3.1　翻訳規則

本稿で用いる翻訳規則 $r(\in \mathcal{R})$ は (t, s) から成り，

- t は翻訳規則の左辺と呼び，係り受けの部分木に相当する．部分木の内部頂点は終端記号 (単語)，それ以外の頂点は frontier であり終端記号か変数をとる (変数の集合 $\mathcal{X} = \{x_1, x_2, \dots\}$)
- s は翻訳規則の右辺と呼び，終端記号と変数の文字列 $s \in (\mathcal{X} \cup \Delta)^*$ からなる

と定義される．具体的な例としては次のような翻訳規則が挙げられる．



⇒ $x1$ は $x2$ 少女 を 見た

図 2: 翻訳規則の例

この例の左辺では内部頂点として saw，girl という単語を持ち、frontier として $x1$, a, $x2$ という変数と単語を持つ．右辺では日本語の単語と変数からなる文字列を持っている．

木構造から文字列への翻訳を行うための翻訳規則抽出法としては一般に GHKM アルゴリズム [3, 2] が知られている．それを森に基づいた規則抽出に拡張した手法も提案されている [10, 9]．本稿ではこの森による GHKM アルゴリズムを係り受け森に適用した．

アルゴリズムでは最小規則と呼ばれる最小単位の取り出し可能な規則を取り出した後，それらを結合することでより大きな単位の規則を作り出す．この操作によって作り出される規則は結合規則と呼ばれる．この結合規則を取り出す際には規則のサイズや規則数の閾値などを設けることで取り出す規則の大きさや数に制限が設けられる．本稿では結合規則の数を各頂点で 10 個に制限した．

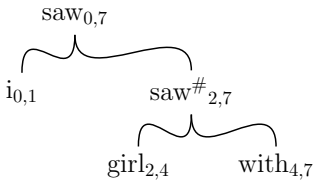


図 3: Binarization 後の係り受けの一部

3.2　翻訳規則の確率値推定

翻訳規則 r から以下の 3 つの確率値を推定する．

$$\phi(r|lhs(r)) = \frac{c(r)}{\sum_{r':lhs(r')=lhs(r)} c(r')} \quad (3)$$

$$\phi(r|rhs(r)) = \frac{c(r)}{\sum_{r':rhs(r')=rhs(r)} c(r')} \quad (4)$$

$$\phi(r|root(lhs(r))) = \frac{c(r)}{\sum_{r':root(lhs(r'))=root(lhs(r))} c(r')} \quad (5)$$

c(r) は翻訳規則 r の出現頻度であるが，本稿では単純に出現を 1 回と数えるのではなく，内側外側アルゴリズムを用いて係り受け解析における尤度最大化に基づいた頻度の期待値を計算する．

頂点 v に対する外側確率を $\alpha(v)$，内側確率を $\beta(v)$ とすると，ある翻訳規則 r に対する内側外側確率 $\alpha\beta(v)$ は

$$\alpha\beta(r) = \alpha(root(r)) \times \prod_{e \in r} p(e) \times \prod_{v' \in leaves(r)} \beta(v') \quad (6)$$

として計算できる．これより翻訳規則 r における出現頻度の期待値は

$$c(r) = \frac{\alpha\beta(r)}{\alpha\beta(TOP)} \quad (7)$$

となる．ここで TOP は係り受け森上の $root$ を示している．

3.3　Binarization

提案システムでは係り受け森から翻訳規則の抽出やデコードを行うことになるが，ある超辺が持つ $tail$ の数が多くなると規則の質やデコードの効率に問題が生じる．そこで本稿では係り受け森を binarization する手法を提案することで超辺が持つ $tail$ の数に制限を加える．ここで binarization とは超辺の $tail$ のサイズが 2 以下であるようにすることである．

本稿で提案する binarization はある超辺が持つ $tail$ のサイズが 3 以上である場合，最も左の要素とその次の要素を

$head$ から生成する擬似的な頂点を $head$ とした新たな超辺の $tail$ として構成し直すという作業を再帰的に行うことである．例として図 1(c) における超辺 $e1$ に対してこの操作を行うと，図 3 のように $saw^{\#}_{2,7}$ を擬似的な $head$ とした新たな超辺が再構成されることになる．係り受け森の $root$ からトップダウンに各頂点を訪れてこの操作を行うことで，係り受け森を binarization することができる．この binarization 手法は Wang07[16] で提案された句構造木の binarization 手法とほぼ同様の手続きである．本稿の実験では binarization をした場合としない場合の両方で翻訳精度の検証を行った．

3.4 対数線形モデル

統計的機械翻訳では対数線形モデルを用いることが一般的である．対数線形モデルは複数の素性を重みで線形結合した形で表される．素性に対する重みは与えられた訓練データ上で翻訳の評価尺度 (BLEU など) を最大化するように学習される．この学習法は誤り率最小化学習法 (MERT) と呼ばれている [12]．

3.5 探索手法

提案するシステムでは係り受け森 F(実際には森の枝刈りを行う) を入力とする．探索ではまず Algorithm1 によって，係り受け森 F の各頂点へボトムアップに訪れながら，それぞれの頂点において適用できる翻訳規則の候補を規則集合 R から探す．適用できる翻訳規則があった場合，現在見ている係り受け森の頂点を $head$，翻訳規則の左辺における変数部分に対応する頂点を $tail$ とした超辺を作成する．ただし，この超辺には翻訳規則の右辺を付与する．具体的な例としては図 2 で示した翻訳規則を図 1(c) の頂点 $\mathrm{saw}_{0,7}$ を $head$ とする超辺 $e2$ に適用した場合，次のような超辺を作る．

$$\langle (\mathrm{i}_{0,1}, \mathrm{with}_{4,7}), \mathrm{saw}_{0,7}, x1 \text{ は } x2 \text{ 少女 を 見た} \rangle$$

また，翻訳に用いる右辺の変数と tail となる頂点の対応関係は翻訳時に必要となるため記憶しておく．全てのノードを訪れた後，係り受け森の全頂点と Algorithm1 によって作り出された超辺からなる超グラフを翻訳森と呼ぶ．

Algorithm 1: 上昇型変換アルゴリズム

```
procedure BottomUpConversion(F, R)
    for each node v ∈ V in (bottom-up) topological order do
        for rule r ∈ R do
            vars ← match(r, v)
        if vars is not empty then
            e ← ⟨vars, v, s(r)⟩
            add translation hyperedge e to H
```

探索ではこの翻訳森の頂点をボトムアップに訪れながら，超辺に付与した翻訳規則の右辺を使って翻訳候補を作り出す．この際，作り出される仮説は膨大な数に及ぶため，ビームサーチを用いる．また，部分的な翻訳候補を組み合わせる際に lazy な操作を行うことで探索の効率化を行っている．ビームサーチと lazy な操作を行いながら探索する手法は cube pruning と呼ばれている [1]．

実際の動作ではこの cube pruning を 1-best の翻訳結果を得るために用いている．モデルの訓練を行うための K-best 出力では cube pruning を全頂点で 1-best を見つけるために動作させた後，Huang05[6] で提案されているアルゴリズム 3 によって高速に K-best を探索を行っている．

表 2: 係り受け解析精度 (unlabeled accuracy)

モデル ＼ テスト	新聞 (chtb)	報道 (ctv)
news	83.56	81.04
all	82.32	82.49

3.6 疑似的な超辺の生成

係り受け森を翻訳森へと変換する際，ある頂点に適用できる翻訳規則がない場合，疑似的な翻訳森の超辺を生成する．例として図 1(c) における頂点 $\mathrm{saw}_{0,7}$ の超辺 $e1$ に対して，疑似的な翻訳規則を生成することを考える．この場合，saw という単語に対する訳語を GIZA++で学習した翻訳 table から辞書引きし [17]，“見た”という単語を得たとすると，

$$\langle (\mathrm{i}_{0,1}, \mathrm{girl}_{2,4}, \mathrm{with}_{4,7}), \mathrm{saw}_{0,7}, x1 \text{ 見た } x2\ x3 \rangle$$

という超辺を作り出す．

しかし，この規則では出力側言語の並びが考慮されておらず，翻訳精度の悪化をまねく危険性がある．そのため，本稿では binarization を行った場合にのみ，変数 (binarization 時は多くても 2 つ) と訳語の並べ方最大 6 通りを考慮して翻訳森の超辺を生成した．デコードは前述したように lazy な操作 (と Huang05 の algorithm2 における超辺の一括処理) を行うため，最大で 6 つの規則を生成したとしても効率良く動作する．binarization を行っていない場合，出力言語側のあらゆる並びを考慮することは超辺の $tail$ サイズが大きくなることがあるため現実的に実行が困難である．

4 実験

本稿では中国語から英語への翻訳で実験を行った．対訳コーパスには新聞データから作成された FBIS コーパスを用いた．中英それぞれ単語数 3.5M，4.3M である．翻訳モデルの学習には GIZA++を用いた．また，言語モデル学習用のデータとして english giga word の xinhua 部分のデータ (単語数 350M) を利用した．N-gram の学習は srilm を用いて 4-gram 言語モデルを学習した．翻訳実験では NIST 2002/2003 Open Machine Translation Evaluation の中英翻訳データを用意し，2002 を MERT，2003 をテストデータとして用いた．中国語の形態素解析には stanford segmenter と stanford postagger を利用した．英語のトークナイゼイションには英語ツリーバンクの処理に使われたツールを用いた．中国語の係り受け解析は 1-st order の Eisner アルゴリズムによって行った．

4.1 係り受け解析実験

ここでは中国語ツリーバンクの全 18,472 文から ctv(報道) 部分 500 文と chtb(新聞) 部分 500 文を除いたデータ (all) と chtb 部分 8981 文から共通の chtb 部分 500 文を除いたデータ (news) とを用いて 2 つの係り受け解析モデルを構築した (係り受けフォーマットへは Penn2Malt を用いて変換)．除いたデータはテストデータとして利用した．

表 2 では係り受け解析の精度 (unlabeled accuracy) を示している．この結果からは新聞データをドメインとする解析を行う場合，新聞データのみから学習したモデルを用いる方が良いことがわかる．よって以下の翻訳実験では新聞をドメインとするデータを用いるため，全新聞データ 8981 文から学習したモデルを用いた．

表 3: GHKM アルゴリズムから抽出した翻訳規則 (binarize なし/あり)

K-best	1	5	10
翻訳規則の数	5.6M/15.6M	7.8M/26.1M	9.6M/32.0M
変数の数	4.4/2.6	4.3/2.5	4.4/2.5

表 4: 提案システムと様々なシステムの性能比較

評価 / システム	BLEU
moses	22.15
joshua	21.84
1-best dep (binarize なし/あり)	18.42/18.38
20-best dep (binarize なし/あり)	21.67/21.82

4.2 翻訳実験

提案するシステムで用いる素性は両方向からの単語翻訳確率，翻訳規則に関する確率 (式 (3),(4),(5))，係り受け解析スコア，N-gram 言語モデル，単語数，規則数の計 9 つである．これらの素性に対する対数線形モデルの重みは MERT で BLEU を最大化するように学習した．

表 3 は提案システムにおける係り受け森の K-best 数と翻訳規則数の関係を示している．また，1 つの翻訳規則が持つ平均の変数の数も示している．翻訳規則が持つ変数の数は少ない方が効率良くデコードを行うことができる．結果からは binarization には変数の数を減らす効果があると言えるが，その一方で規則の数は大きく増加した．以下の翻訳実験ではテストデータや訓練データに対して使われることのない規則は削除したが，獲得される規則の数を減らす手法は今後の課題である．

提案システムとの比較には句に基づくデコーダ moses と階層句に基づくデコーダ joshua を用いた．対数線形モデルの重み訓練時の K-best 数は各システム共に 2000-best 出力した．またデコード時の翻訳候補 K-best 数は各システム共に 500-best とした．表 4 に BLEU による翻訳精度の差を示す．ここでは 10-best の係り受け森を用いて抽出した翻訳規則を使って翻訳実験を行った．実験結果から提案システムは階層句や句に基づいたシステムとほぼ同等の性能を示すことができたが，翻訳精度の著しい向上は得られなかった．

この原因として係り受けは単語で超グラフの頂点が表されるため，翻訳規則の適用を行う際，未知語などの影響を大きく受け易いことが挙げられる．本稿では森によるアプローチと binarization によってこの問題を緩和させたが，今後はより柔軟な規則適用が行える係り受け文法を検討する必要があると言える．

5 まとめ

本稿では係り受け森を入力とする統計的機械翻訳システムを提案した．中国語から英語への翻訳実験からは従来のオープンソースシステムとほぼ同等の性能を示すことができた．今後は日本語などより係り受け解析に適した言語を対象とした翻訳実験を行う予定である．

システムの改善としては 4.2 節でも述べたように，柔軟な規則適用ができる文法形式を模索する必要があると言える．また，本稿では入力側に係り受け森を用いただけであったが，出力側にも係り受け森を利用したシステムへと改善することも検討している．

参考文献

[1] D. Chiang. Hierachical phrase-based translation. *Computational Linguitics*, 33:201–228, 2007.

[2] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of Coling-ACL*, pages 961–968, July 2006.

[3] M. Galley, M. Hopkins, K. Knight, and D. Marcu. What's in a translation rule? In *HLT-NAACL*, pages 273–280, May 2004.

[4] K. Gimpel and N. A. Smith. Feature-rich translation by quasi-synchronous lattice parsing. In *Proceedings of EMNLP*, pages 219–228, August 2009.

[5] J. Graehl and K. Knight. Training tree transducers. In *HLT-NAACL*, pages 105–112, May 2004.

[6] L. Huang and D. Chiang. Better k-best parsing. In *Proceedings of IWPT*, pages 53–64, October 2005.

[7] L. Huang, K. Knight, and A. Joshi. A syntax-directed translator with extended domain of locality. In *Proceedings of AAMT*, pages 1–8, June 2006.

[8] Y. Kenji and K. Kevin. A syntax-based statistical translation model. In *Proceedings of ACL*, pages 523–530, July 2001.

[9] Y. Liu, Y. Lü, and Q. Liu. Improving tree-to-tree translation with packed forests. In *Proceedings of ACL*, pages 558–566, August 2009.

[10] H. Mi and L. Huang. Forest-based translation rule extraction. In *Proceedings of EMNLP*, pages 206–214, October 2008.

[11] H. Mi and Q. Liu. Constituency to dependency translation with forests. In *Proceedings of ACL*, pages 1433–1442, July 2010.

[12] F. J. Och. Minimum error rate training in statistical machine translation. In *Proc. the 41th ACL*, pages 160–167, 2003.

[13] L. Shen, J. Xu, and R. Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-HLT*, pages 577–585, June 2008.

[14] L. Shen, J. Xu, B. Zhang, S. Matsoukas, and R. Weischedel. Effective use of linguistic and contextual information for statistical machine translation. In *Proceedings of EMNLP*, pages 72–80, August 2009.

[15] Z. Tu, Y. Liu, Y. S. Hwang, Q. Liu, and S. Lin. Dependency forest for statistical machine translation. In *Proceedings of Coling*, pages 1092–1100, August 2010.

[16] W. Wang, K. Knight, and D. Marcu. Binarizing syntax trees to improve syntax-based machine translation accuracy. In *Proceedings of EMNLP-CoNLL*, pages 746–754, June 2007.

[17] D. Yuan and P. Martha. Machine translation using probabilistic synchronous dependency insertion grammars. In *ACL*, 2005.