パラレルデータの階層的フレーズアラインメント

後藤 功雄　　田中 英輝
NHK 放送技術研究所

1　はじめに

日本語のニュースを英語へ自動翻訳する研究を行っている．この自動翻訳では，日英対訳の表現対（翻訳部品）を必要とする．特に，対訳関係にある任意長の表現（フレーズ）対からなる翻訳部品が，自動翻訳に有用である．そのため，日英の対訳文データ（パラレルデータ）中におけるフレーズの対訳関係を推定（アラインメント）することにより，フレーズ単位の翻訳部品を獲得することを目指している．

従来の単語単位でアラインメントする手法（e.g. GIZA++[1]）は，複数の単語からなる表現同士をアラインメントすることができない．

本稿では，パラレルデータの階層的なフレーズアラインメント手法を提案する．提案手法は，アラインメントするフレーズの長さに制限がなく，アラインメント結果は階層的で日英対称になるという特徴がある．

以下，2 章で提案手法について説明し，3 章で実験について述べ，4 章でまとめる．

2　提案手法

まず，提案手法の概要について説明する．提案手法は，次の２段階の処理でアラインメントする．

1. パラレルデータから複数の統計量がしきい値以上となる対訳フレーズ候補を抽出する．
2. 他のアラインメントと階層的に整合性がとれる対訳フレーズ候補の中から，複数の特徴量を用いて識別的に候補を選択していくことでアラインメントする．特徴量には，単語レベルとフレーズレベルの統計量や対訳辞書の登録の有無を利用する．

以下，１段目の処理である対訳フレーズ候補の抽出，２段目の処理である階層的なアラインメント，さらに，スコアの計算で利用するパラメータの推定方法について述べる．

2.1　対訳フレーズ候補の抽出

本節では，フレーズ対の列挙アルゴリズムと枝刈り手法を用いて，複数の統計量がしきい値以上となるフレーズ対を効率的に抽出する手法について述べる．ここでフレーズは，連続する１つ以上の単語からなる任意長の表現とする．

日英各言語毎に頻出表現を抽出すると，取得される表現の数が多くなるため，その日英の表現の組合せ数は膨大になり，組合せの計算はデータ量が多いと困難になる．

ただし，共起する文数がしきい値以上の表現対を探索する場合に，各言語毎に出現文数がしきい値以上の表現を抽出して，得られた日本語表現と英語表現の組が共起する文を数えるとしきい値未満になる組が多く含まれると考えられる．

そこで，提案手法では，共起する文数がしきい値以上となるフレーズ対を直接探索する．提案手法のアルゴリズムの基本的な考え方は，深さ優先探索を２段階で行い，頻度以外の統計量を用いて枝刈りするというものである．

以下，まず，単言語での出現文数が多いフレーズの深さ優先探索を定式化し，次に，提案手法について説明する．

2.1.1　単言語でのフレーズの深さ優先探索手法

深さ優先探索を利用した単言語での出現文数が多いフレーズの探索を以下のように定式化する．

単語の集合を $W = \{w_1, w_2, \cdots, w_n\}$ とする．文を s，文番号を d とし，文番号と文のペア (d, s) の集合をコーパス S とする．フレーズを p とする．単語の系列を $\boldsymbol{f} = f_1 f_2 \cdots f_l$；$f_i \in W$，$i \in \{1, 2, \cdots, l\}$ と定義する．s と p は系列で表される．フレーズ p のコーパス S 中での出現文数を $c(p)$ とする．

単言語での出現文数が ζ 以上のフレーズの探索とは，任意の自然数 ζ に対し，$c(p) \geq \zeta$ となるフレーズ p をすべて列挙することとする．

これは，以下に示す深さ優先探索に基づくアルゴリズムにより実行することができる．

ここでアルゴリズムの説明の前に，以下の変数を定義する．ある系列 $\boldsymbol{f}$ を含むコーパスを $X \subseteq S$ とする．索引として利用する集合 Y，H，G を次のように定義する．X において，文番号 d と，d の文中で $\boldsymbol{f}$ に一致する末尾の単語位置 +1 の値 r とのペア (d, r) の集合を Y とする．ただし，r が文末の単語位置より大きい場合は Y に含めない．X 中の全ての (d, s) において，$(d, r) \in Y$ の d と r と単語位置が r の単語 g との組 (d, r, g) の集合を H とする．H に含まれる単語の集合を G とする．

図 1 に，$\zeta = 2$ とした場合で，a で始まるフレーズを取得する動作例を示す．

1. f_1 = a とし，$\boldsymbol{f} = f_1$ を含む文番号と文のペア集合 X と，文番号と a が出現した位置とのペアの集合 Y を生成する．
2. X の基数 $|X|$ が 2 以上の場合，$\boldsymbol{f}$ をフレーズとして出力し，次の処理に進む．2 未満の場合，処理は終了する．
3. X と Y から H と G を生成する．$\boldsymbol{f}$ に後続する各単語 $w_i \in G$ について，$\boldsymbol{f}$ の末尾に w_i を追加した $\bar{\boldsymbol{f}}$，$\bar{X} = \{(d, s) | ((d, s) \in X) \wedge ((d, r, g) \in H) \wedge (g = w_i)\}$，$\bar{Y} = \{(d, r) | ((d, r, g) \in H) \wedge (g = w_i)\}$ を作成する．$\bar{\boldsymbol{f}}$，$\bar{X}$，$\bar{Y}$ を新たな $\boldsymbol{f}$，X，Y として，上記 2. の処理に戻り，以後再帰的にこれらの処理を繰り返すことで a から始まる全ての $c(p) \geq 2$ となるフレーズを抽出できる．

疑似コードを図 2 に示す．ここで Y_{init} とは，$(d, s) \in S$ における d と s 中の全ての単語の位置-1 とのペアの集合とする．

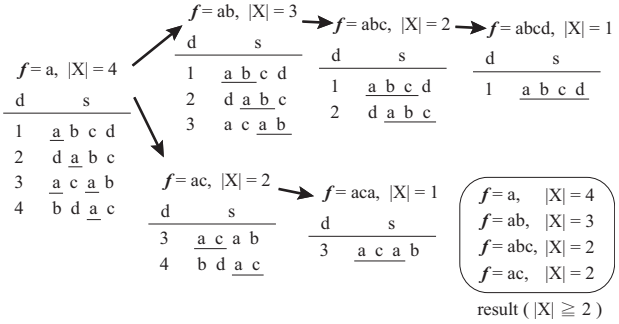

図 1　単言語での頻出フレーズの深さ優先探索の動作例

```
1  call DepthFirstSearch( φ, S, Y_init )

2  procedure DepthFirstSearch( f, X, Y )
3  begin
4    make H and G from X and Y
5    foreach w_i ∈ G do
6      f̄ ← f with w_i
7      X̄ ← {(d, s)|((d, s) ∈ X) ∧ ((d, r, g) ∈ H)
                                        ∧(g = w_i)}
8      Ȳ ← {(d, r)|((d, r, g) ∈ H) ∧ (g = w_i)}
9      if |X̄| ≥ ζ then
10       output f̄ as a phrase
11       call DepthFirstSearch( f̄, X̄, Ȳ )
12     endif
13   end
14 end
```

図 2　フレーズの深さ優先探索アルゴリズム

```
1  call ExpandJ( φ, S, Y^j_init )

2  procedure ExpandJ( f^j, X, Y^j )
3  begin
4    make H^j and G^j from X and Y^j
5    foreach w^j_i ∈ G^j do
6      f̄^j ← f^j with w^j_i
7      X̄ ← {(d, s^j, s^e)|((d, s^j, s^e) ∈ X) ∧ ((d, r^j, g^j) ∈ H^j)
                                        ∧(g^j = w^j_i)}
8      Ȳ^j ← {(d, r^j)|((d, r^j, g^j) ∈ H^j) ∧ (g^j = w^j_i)}
9      if |X̄| ≥ ζ then
10       Ȳ^e ← {(d, r^e)|((d, r^e) ∈ Y^e_init) ∧ ((d, s^j, s^e) ∈ X̄)}
11       call ExpandE( φ, X̄, Ȳ^e, f̄^j )
12       call ExpandJ( f^j, X̄, Ȳ^j )
13     endif
14   end
15 end

16 procedure ExpandE( f^e, X, Y^e, f^j )
17 begin
18   make H^e and G^e from X and Y^e
19   foreach w^e_i ∈ G^e do
20     f̄^e ← f^e with w^e_i
21     X̄ ← {(d, s^j, s^e)|((d, s^j, s^e) ∈ X) ∧ ((d, r^e, g^e) ∈ H^e)
                                        ∧(g^e = w^e_i)}
22     Ȳ^e ← {(d, r^e)|((d, r^e, g^e) ∈ H^e) ∧ (g^e = w^e_i)}
23     if |X̄| ≥ ζ then
24       output f^j and f̄^e as a phrase pair
25       call ExpandE( f̄^e, X̄, Ȳ^e, f^j )
26     endif
27   end
28 end
```

図 3　フレーズ対の列挙アルゴリズム

2.1.2　提案する対訳フレーズ候補の抽出手法

提案する対訳フレーズ候補の抽出手法について説明する．まず共起文数に基づくフレーズ対の列挙アルゴリズムについて述べ，次に対訳らしさの統計的指標について述べ，最後に統計量に基づく枝刈り手法について述べる．

共起文数に基づくフレーズ対の列挙アルゴリズム

まず，パラレルデータの各言語を区別するために前節で導入した変数を拡張する．前節で導入した変数に，一方の言語（ここでは日本語とする）についての変数には，変数の右肩に j を，もう一方の言語（ここでは英語とする）についての変数には，変数の右肩に e を付与する．例えば，s^j は，日本語の文を示し，s^e は英語の文を示す．また，コーパス S および X は，文番号 d と日本語文 s^j と英語文 s^e との組 (d, s^j, e^j) とする．なお，パラレルデータは対訳関係にある日英の文が文番号を共有するため，文番号に言語の区別はない．

表 1　2×2 分割表

	$\boldsymbol{f}^j$	$\neg\boldsymbol{f}^j$
$\boldsymbol{f}^e$	a_1	a_2
$\neg\boldsymbol{f}^e$	a_3	a_4

提案する深さ優先探索に基づく頻出フレーズ対の抽出手法の疑似コードを図 3 に示す．単言語でフレーズを深さ優先探索する ExpandJ の中で，もう一方の言語のフレーズを深さ優先探索する ExpandE を呼び出し（図 3 の 11 行目）て，深さ優先探索を 2 段階で行っている．

このアルゴリズムには，次の特徴がある．図 3 の 11 行目で 2 段目の探索を行う ExpandE を呼び出す際に，引数として与えるコーパスは S ではなく $\bar{X}$ とすることによって，ExpandE は，$\boldsymbol{f}^j$ が出現する文のみを探索することになる．これによって，図 3 の 23 行目の $|\bar{X}|$ は，$\boldsymbol{f}^j$ と $\bar{\boldsymbol{f}}^e$ のフレーズ対が共起する文数となり，共起文数がしきい値 ζ 以上のフレーズ対を直接列挙できる．

対訳らしさの統計的指標

前記の共起文数に基づくフレーズ対の列挙アルゴリズムにより列挙されるフレーズ対には，対訳である可能性が低いものも多く含まれる．そこで，その中から対訳である可能性が高いものを対訳フレーズ候補として統計的に選択する．

対訳らしさを示す統計的な指標として，以下に示す 4 つの統計量を用いる．これらがしきい値以上となるフレーズ対を対訳フレーズ候補として取得する．また，これらの統計量も取得しておき，次のアラインメントの処理で利用する．

以下，本稿において，有意確率，Dice 係数，フレーズ平均生成確率，フレーズ生成確率と呼ぶ統計量を説明する．

- 有意確率
  統計的仮説検定である Fisher's exact test の片側検定をフレーズ対の共起文数について行い，その有意確率（p-value）を 2 倍して負の対数をとった値とする．つまり，$-\log(\text{p-value} \times 2)$．ここでコーパス中のデータ数（文数）$a_1$，$a_2$，$a_3$，$a_4$ を表 1 の 2×2 分割表のように定義する．すると p-value は，次式で計算することができる．

$$\text{p-value} = b + \sum_{i=1}^{\min(a_2,a_3)} \frac{(a_3 - i + 1)(a_2 - i + 1)b}{(a_1 + i)(a_4 + i)}$$

$$b = \frac{(a_1 + a_2)!(a_3 + a_4)!(a_1 + a_3)!(a_2 + a_4)!}{a_1!a_2!a_3!a_4!(a_1 + a_2 + a_3 + a_4)!}$$

  なお，$a_1 + a_2$ の値は，図 3 の 11 行目で ExpandE を呼び出す際に引数として X の他に S も与えて，この引数 S についても，X と同様の操作を行うことで取得できる．
- Dice 係数
  Dice 係数の値．表 1 の値を用いて次式で計算する．

$$\frac{2a_1}{(a_1 + a_3) + (a_1 + a_2)}$$

- フレーズ平均生成確率
  次式で定義する値．単語の条件付き確率 P は，EM アルゴリズムにより最尤推定する確率モデルである IBM

model 1[1] を用いる．

$$\left(\frac{1}{l^j}\sum_{i=1}^{l^j}\max_k P(f_i^j|f_k^e)\right)^{1/2}\left(\frac{1}{l^e}\sum_{i=1}^{l^e}\max_k P(f_i^e|f_k^j)\right)^{1/2}$$

- フレーズ生成確率
  次式で定義する値．単語の条件付き確率 P は，IBM model 1 を用いる．

$$\max\left(\frac{1}{l^j}\sum_{i=1}^{l^j}\frac{1}{l^e}\sum_{k=1}^{l^e}P(f_i^j|f_k^e),\frac{1}{l^e}\sum_{i=1}^{l^e}\frac{1}{l^j}\sum_{k=1}^{l^j}P(f_i^e|f_k^j)\right)$$

統計量に基づく枝刈り手法

計算量を削減するために，探索空間を枝刈りする手法について述べる．

共起文数が ζ 以上のフレーズ対のうち，前記の統計量がしきい値未満となるものを探索空間から除けば探索の計算量を削減できる．しかし，前記 4 つの統計量において，しきい値以上となるフレーズ対を直接列挙することは困難である．

そこで，探索中にこれから探索する範囲に統計量がしきい値以上となるフレーズ対が出現するかどうかを予測して，出現する見込みが少ないと予測した範囲を探索空間から除いて枝刈りする．具体的には次のようにする．フレーズ $\boldsymbol{f}^e$ の統計量がしきい値未満の時に，図 3 の ExpandE で $\boldsymbol{f}^e$ を延長していく場合を考える．$-\log(\text{p-value}\times 2)$ と Dice 係数の値が $\boldsymbol{f}^e$ より 1 単語長い $\bar{\boldsymbol{f}}^e$ の方が小さくなった場合に，それ以上延長してもしきい値以上にならないと予測して延長を止める．

この予測は，次の考え方に基づいている．p^j の対訳の p^e が長い表現である場合，p^e の先頭からの部分的な系列 $\boldsymbol{f}^e$ は，長さが対訳の p^e に近づくほど統計量が大きくなることが期待される．つまり，統計量がしきい値未満の時，$\boldsymbol{f}^e$ を延長しても統計量が小さくなる場合は，それ以上延長しても統計量がしきい値以上にならないことが期待される．

2.2 階層的なフレーズ対のアラインメント

本節では，階層的なフレーズ対のアラインメントを識別的に行う手法について述べる．まずアラインメントの方法について述べ，次にアラインメントで利用する統計的な特徴量について述べ，その後スコアの計算方法について述べ，最後に位置選択手法について述べる．

アラインメントの方法

対訳の文ペアに出現する対訳フレーズ候補の中から，対訳フレーズを識別的に選択することでアラインメントする．ここでアラインメントするフレーズ対は，他のアラインメントと重複しないものまたは階層的な関係にあるものに制限する．この制限を満たすアラインメントを整合するアラインメントと呼ぶ．アラインメントは，既に確定したアラインメントと整合する対訳フレーズ候補のうち，最も対訳らしい候補を選択していくことで行う．このように階層的にアラインメントすることによって，対訳である可能性が高いアラインメントを根拠として，新たな対訳フレーズを決定することができる．

対訳らしさの順位は次のように決定する．まず対訳辞書に登録があるもの，次にフレーズ対の両方に内容語類を含む候補で統計的な特徴量に基づくスコアが大きいもの，次に残りの候補で統計的な特徴量に基づくスコアが大きいものの順とする．アラインメントとして統計的に選択する対訳フレーズ候補は，スコアがしきい値以上のものとする．

統計的な特徴量

対訳らしさのスコアを計算する際に用いる特徴量を示す．はじめの 4 つは 2.1.2 節で説明した統計量である．

- 有意確率
- Dice 係数
- フレーズ平均生成確率
- フレーズ生成確率
- 単語アラインメント結果含有率
  単語単位のアラインメント結果において，対訳フレーズ候補のフレーズ中の単語のうち，対応するフレーズ中の単語にアラインメントされた単語の率．ここでは，単語単位のアラインメント結果として GIZA++ の標準設定である IBM model4 のアラインメント結果を用いる．単語アラインメントを日英，英日の双方向について行った結果の AND と OR を計算し，AND と OR それぞれについての単語アラインメント結果含有率を用いる．
- 対訳フレーズ候補アラインメント率
  パラレルデータをアラインメントした結果において，対訳フレーズ候補がアラインメントとして選択された数 q_s と対訳フレーズ候補が出現した総数 q_a を用いて，$(q_s+1)/(q_a+1)$ と定義する．

スコアの計算手法

前記の特徴量のうち，$-\log(\text{p-value}\times 2)$ の値を h_1 とし，それ以外の特徴量の値を $h_2, h_3, \cdots, h_7$ とする．スコア（score）は次式で定義する．

$$\text{score} = \lambda_1 \log(h_1) + \sum_{i=2}^{7}\lambda_i \log\frac{h_i+\gamma_i}{\gamma_i}$$

ここで，λ_i は各特徴量を重みづけるパラメータ，γ_i はスムージングのパラメータである．

位置選択手法

同じフレーズが文中の複数位置に出現する場合は，どの位置のフレーズをアラインメントするかを統計的に選択する．選択は，周囲のフレーズ対の Dice 係数と，周囲の確定しているアラインメントとの相対距離の確率を利用して行う．

2.3 パラメータの推定

本節では，まず本稿で recall と呼ぶ値の計算方法について述べ，次に recall を用いて前節のスコアで利用するパラメータを推定する手法について述べる．

recall

最小単位で正解のアラインメントが付与された正解データを用いる．正解データのアラインメントは階層的ではない．フレーズ対のアラインメントにおいて，フレーズ中の単語は対応するフレーズ中の全ての単語とアラインメントされていると見なす．

ここで，以下の変数を定義する．正解データにおいてアラインメントされた単語位置の集合を R^j（日本語），R^e（英語）とする．正解データにおいて日本語単語位置 r の単語とアラインメントされた単語数を u_r^j とする（英語の u_r^e も同様）．自動アラインメント結果において，日本語単語位置 r の単語とアラインメントされた単語のうち，正解と一致する数を v_r^j とする（英語の v_r^e も同様）．1 文中の正解の対訳フレーズ対の数を w とする．自動アラインメント結果の各対訳フレーズ対

表 2　実験に用いたデータ

種別	データ数（記事数）
辞書用データ	415　(80)
開発データ	46　(10)
テストデータ 1	52　(10)
テストデータ 2	253　(50)
テストデータ 3	10,127　(1180)

について，重複する正解の対訳フレーズ対の数を m とする．

階層的なアラインメントの中で recall の計算に用いるアラインメントは，日本語の単語位置 r について計算する場合は，日本語フレーズで r の位置の単語を含む最小のものとし，そのフレーズに対応する英語フレーズは最大のものとする（英語の単語位置の場合も同様）．

本稿では，以下の式で定義する値を recall と呼ぶ．

$$\text{recall} = \frac{1}{|R^j| + |R^e|}\left(\sum_{r \in R^j} \frac{v_r^j}{u_r^j}\frac{(w-m)}{(w-1)} + \sum_{r \in R^e} \frac{v_r^e}{u_r^e}\frac{(w-m)}{(w-1)}\right)$$

ここで，v_r^j/u_r^j と v_r^e/u_r^e は単語単位での正解アラインメントカバー率である．$(w-m)/(w-1)$ はアラインメントの解像度に関する項で，短いフレーズでアラインメントできずに長いフレーズのみでアラインメントされている場合にペナルティを与えるものである．

パラメータ推定手法

エラー率 = 1 – recall と定義し，開発データを用いてエラー率最小化学習 [2] を行うことで，パラメータを推定する．

3　実験

3.1　実験設定

日英の文対応がついている NHK 気象・災害ニュースをアラインメントする実験を行った．実験に用いたデータの種類と数を表 2 に示す．テストデータ 3 以外のデータは，最小単位で正解のアラインメントが付与された 150 記事分のデータを記事単位でランダムに分割して作成した．

開発データとテストデータ 1 と 2 と 3 の計 10,478 データをアラインメント用データとした．このデータに対して対訳フレーズ候補の抽出とアラインメントを行った．抽出時のしきい値は，$\zeta = 5$，$-\log(\text{p-value} \times 2) = 8.56$，Dice 係数 = 0.05，フレーズ平均生成確率 = 0，フレーズ生成確率 = 0.001 とした．ただし，対訳候補が 1 単語同士の場合は $\zeta = 1$ とし，しきい値に フレーズ生成確率 = 0.001 のみを用いた．辞書用データのアラインメントを対訳辞書として利用した．開発データを用いてパラメータの推定を行った．

3.2　実験結果と考察

図 4 にテストデータ 1 のアラインメント結果においてスコアのしきい値を変化させたときの対訳辞書に登録がないアラインメントの数と，その precision（statistical alignment）及び全体の precision（all alignment）との関係を示す．スコアが高いものは対訳である可能性が高いことが分かる．

表 3 にアラインメント用データのアラインメント結果で 4 単語以上のフレーズのものを頻度でランキングした N-best を示す（上位のものとフレーズのどちらかが同じもの，同形の数値表現，対訳辞書にあるものは除く）．意味的にある程度まとまりのある単位の翻訳部品が取得できていることが分かる．

表 4 にテストデータ 1 と 2 の計 305 データについての複数の手法の recall を示す．提案手法はスコアのしきい値を 0 とした場合の結果である．提案手法（辞書無し）は，提案手法で対訳辞書を用いなかった場合，model4 は GIZA++ を標準設定で用いた場合である．提案手法は，GIZA++ の片方向（JE, EJ）のアラインメントより値が高くなっており，これらより幅広くアラインメントできている．OR は，提案手法より値が高くなっている．しかし OR のアラインメントは分散した単語からなる場合があり，この場合は翻訳部品として利用しにくい．それに対して，提案手法でアラインメントされるフレーズは連続しているため，翻訳部品として利用しやすい．

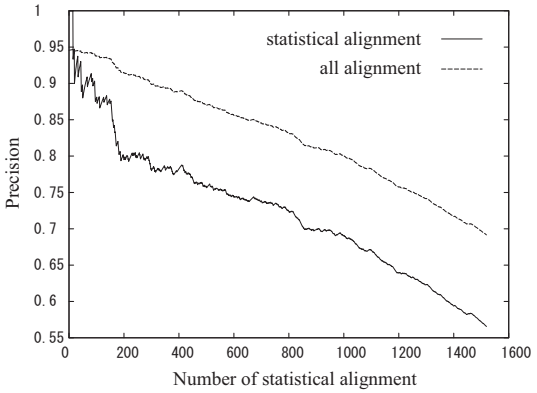

図 4　統計的なアラインメントの N-best 数と precision の関係

表 3　アラインメントされたフレーズ対

日本語フレーズ	英語フレーズ
気象庁 の 発表 に より ます と	The Meteorological Agency says
台風 の 暴風 域 に 入っ	the typhoon 's storm zone
の 最大 風速 は	is packing winds of
に 三十 ミリ から 五十 ミリ	30 to 50 millimeters
四国 の 太平洋 側	the Pacific side of Shikoku
大型 で 強い 台風	large and powerful typhoon
大型 で 非常 に 強い 台風	large and very powerful typhoon
四国 の 瀬戸内 側	Sea side of Shikoku
や 河川 の はんらん	the flooding of rivers
台風 の 北上 に	the typhoon moves north
暴風 域 に 入る	enter its storm zone
過去 数 年間 で	the past several years
四国 の 瀬戸内 側	Inland Sea side of Shikoku
台風 の 北上 に	the typhoon heads north
大型 で 強い 台風	A large and powerful typhoon
中心 付近 の 最大 風速 は 四十	The maximum wind speed near the center is 144
北 北東 に 進ん で いる	is moving north northeast

表 4　recall の比較

提案手法	提案手法	model 4			
	辞書無し	JE	EJ	AND	OR
75.8	71.3	64.5	63.5	49.3	85.4

4　おわりに

パラレルデータの階層的なフレーズアラインメント手法を提案した．提案手法は，アラインメントするフレーズの長さに制限がなく，アラインメント結果は階層的で日英対称になるという特徴がある．NHK 気象・災害ニュースのパラレルデータをアラインメントしてフレーズの翻訳部品を獲得できることを示した．

参考文献

[1] Franz Josef Och, Hermann Ney. “A Systematic Comparison of Various Statistical Alignment Models,” Computational Linguistics, volume 29, number 1, pp.19-51, March 2003.

[2] Franz Josef Och. “Minimum Error Rate Training in Statistical Machine Translation,” ACL, pp.160-167, 2003.