構造を利用した文書間の細粒度対応付け手法

辻尾　尚樹† 　清水　敏之†† 　吉川　正俊††

† 京都大学工学部情報学科　〒 606–8501 京都府京都市左京区吉田本町
†† 京都大学大学院情報学研究科　〒 606–8501 京都府京都市左京区吉田本町
E-mail: †tsujio@db.soc.i.kyoto-u.ac.jp, ††{tshimizu,yoshikawa}@i.kyoto-u.ac.jp

あらまし　同一の話題に対する複数の形態による文書や，バージョン違いの文書，例えば，国際会議において発表された論文とその研究を発展させて執筆された論文誌論文など，それら文書間で互いの部分的な対応関係が把握できることは有用である．しかしそのような文書では，対応する部分が大きく書き変わっており，内容を比較するだけでは適切な対応付けができない場合がある．そこで，本研究では対応付けに文書の内容だけでなく，節構造や段落構成といった文書の構造情報も利用することを考えた．我々は，構造化文書を対象とし，文書間で互いの対応部分を取得するために，文書の内容と構造を考慮した対応付けの手法を提案する．

キーワード　文書アライメント，構造化文書

1. はじめに

国際会議において発表された論文とその研究を発展させて執筆された論文誌論文などといった同一の話題に対する複数の形態による文書や，Wikipedia のある記事の履歴などといったバージョン違いの文書では，一方の文書のある段落が他方の文書のある段落に対応する，またはある節が他方のある節に対応する，というようにそれら文書間で互いに部分的な対応を取ることができる．

文書の各々の部分でそのような対応関係が把握できることは有用である．例えば文書を読んでいて理解し辛い部分があった場合に他方の文書の対応する部分を読むことで理解の助けになったり，過去に読んだ文書からの差分だけを読みたい場合に，文書間の部分対応の情報に基づいて，それらの差分を提供することができる．また，情報量の小さな部分を対応する情報量の大きな部分で補完したり，文書の剽窃の検出に利用することも考えられる．このように，文書間の部分対応付けには様々な応用場面が考えられ，重要な役割を担っている．

本研究では，文書間で各々の部分がどのように対応しているかを把握するために，文書間で細粒度での対応付けを行うことを考える．

我々は，文書内の部分文書にはそれぞれ文書内における役割があると考えた．例えば，論文のような文書を意識すると，各節の文書内での役割として，文書の導入，文書の主となる話題の説明，そして文書のまとめなどが考えられる．節の役割は節のタイトルとして現れている場合もある．さらに，節内の各段落の役割として，節の導入，数式の提示，定義の導入，導入した定義の説明などが考えられる．本研究では，それら文書を構成する要素について，そのような与えられた役割が同じであることを以て「対応する」，対応する要素同士を関連付けることを「対応付ける」と呼ぶことにする．最初に挙げた，同一の話題に対する複数の形態による文書やバージョン違いの文書も，「ある同一の対象について記述する」という役割が共通するという点で,「対応する」文書と言える．

対応する文書間でさらに細粒度で対応付けを行うのであるが，そのような文書では，ある部分は大きく書き変わっていたり，ある部分は削除，追加されていたり，あるいは分割，入れ替わりが起こっていたりする場合がある．また，それらの修正が複数組み合わさっている場合もある．文書のある部分同士が対応するかどうかの判定にはそれらの内容が類似しているかを見るのが一つの基本的な手段であるが，複雑な修正がなされた部分は内容の類似度は低くなるため，内容の情報だけで適切な対応付けを行うのは難しい．

さらに，対応する文書が常に同じ言語で書かれているとは限らない．論文の例で言うと，国内会議にて発表する論文は母国語で書かれており，その後国外の会議にて発表する際には英語で書き直されているだろう．それらを対応付けるには，自動翻訳を利用して論文をどちらかの言語に揃えてから処理するといった方法が考えられるが，人手で翻訳したものと自動翻訳を利用したものとでは単語の選択や翻訳者の表現の癖など，翻訳結果に差があり，やはり内容の類似度だけでは適切な対応付けは難しい．また，論文の内容も加筆，修正がなされていることが多く，その場合にはさらに内容の類似度は低くなる．

そこで，対応付けの材料として文書の内容の情報だけでなく，文書の構造の情報も利用することを考える．例えば，ある対応する段落同士が大きく書き変わっている場合，それらの内容の類似度だけでは対応付けは難しい．しかし，その段落が含まれる節内の他の段落は大きくは書き変わっておらず，内容の類似度だけでも十分に対応が取れる場合には，その情報は目的の段落を対応付けるための有効な手掛かりとなる．あるいは，節の入れ替わりが起こっていて，節内の段落が書き変わっている場合でも，段落や小節などの節の構造の情報を加味することによって入れ替わり先の節を見つけられることが期待できる．また，節のタイトルを用いて節同士を対応付けるというのも構造

を利用した対応付けと言える．このように，たとえ文書間で対応する部分の内容が大きく書き変わっていて内容の情報だけでは対応が取れない場合でも，文書の構造的な情報を利用することによって適切な対応付けが可能になると考えられる．

我々は，構造化文書を対象とし，文書間で互いの対応部分を取得するために，文書の内容と構造を考慮した細粒度での対応付けの手法を提案する．対応付けには，文字列間の距離の尺度であるレーベンシュタイン距離のアルゴリズムを段落列に対して適用することを考える．さらに，複数段落間での対応付けや，段落の順序の入れ替わりも考慮する．

また，対応付けに文書構造を利用することの有効性を確かめるために，評価実験を行った．実験では，対応文書の例として異なる形態で発表された論文を取り上げ，対応付けを行った．その結果，構造を利用することでより適切な対応付けがなされることが確認され，特に言語の異なる文書ペアで構造を利用する対応付けが有効であることが分かった．

以降，2 節で関連研究について述べた後，3 節で提案手法の詳細を説明する．4 節で提案手法の評価実験を行い，考察を述べる．そして 5 節で本稿を総括する．

2. 関 連 研 究

2.1 文書アライメント

文章アライメントの研究は広く行われている．Daumé III ら [1] は，文書とその概要文との短いフレーズレベルでのアライメントを考えており，隠れマルコフモデルを利用した手法を提案している．Yahyaei ら [7] は，異なる言語で書かれた文書間のアライメントを考えている．Jeong ら [2] は，Bayesian model というモデルを利用して文書間のアライメントを考えている．Yeung ら [8] は，Wikipedia のページが言語によって情報量に差があることを問題と考え，情報量の小さなページの内容を異なる言語で書かれたより情報量の大きいページの内容で補完するという目的にアライメントの技術を応用している．山本 [13] は，保険の約款 (契約の内容が正確かつ詳細に記載された文書) と約款を基に消費者向けに作成されたパンフレットや重要事項説明等の文書とのアライメントを考えており，単語の出現頻度情報を利用したり，タイトルや用語の定義部分に出現する単語を特別視するといった手法を提案している．丸川ら [12] は，交差に対応したアライメント手法を提案し，適用例として特許文書の請求項と発明の詳細な説明とのアライメントを挙げている．

我々の目的は文書間の細粒度対応付け，即ちアライメントを取ることであり，これらの研究と大きな目的は同じである．しかし，これらの研究は全て文書の内容の情報を基にアライメントを取る手法を考えており，我々は対応付けの手法に文書の構造の利用を取り入れたという点でこれらと異なる．

2.2 文書間の類似度算出

文書間の類似度算出の研究は，類似文書検索の分野で多く見られる．Zhang ら [10] は，文書全体を表すルートノードとそれを親として持つパラグラフを表す子ノードという 2 層の単純な木構造に文書をモデル化し，Earth Mover's Distance (EMD) という尺度を用いて異なる文書間の類似度を計算している．また，彼らはこれを文書の剽窃の検出にも応用している [9]．Wan [6] は，文書をいくつかのサブトピックの列として捉え，こちらも EMD を用いた類似度算出の手法を提案している．Tekli ら [4] は，XML 文書を対象とし，Tree Edit Distance を利用して構造的な情報も類似度の計算に取り入れている．

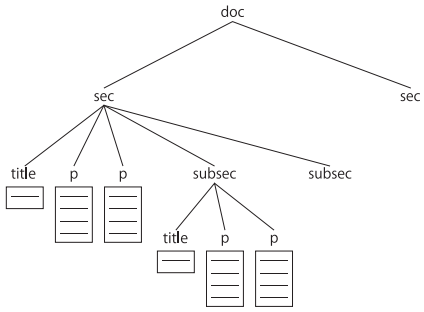

図 1 文 書 木

これらの研究は文書の構造を考慮するというアイデアは我々と同じだが，文書間の類似度算出に焦点を当てており，我々の目的である対応付けとは異なるものである．また，検索への利用を想定しているため，文書を扱うモデルは単純なものにして計算量を減らす工夫が見られるものも多い．

3. 構造を利用した対応付け手法

我々は対応付けの対象を構造化文書としている．構造化文書は内部に節，段落といった構造を持っており，節がさらにいくつかの小節に分けられている場合もある．また，箇条書きも文書の構造として考えることもできる．本研究では，このような構造化文書を木構造として捉え，これを文書木と呼ぶ．図 1 に文書木の例を示す．そして提案手法へは対応付けたい二つの文書木を入力し，対応付けた結果を出力する．対応付けの粒度としてはいわゆる段落を単位とする．節単位や文単位での対応が把握できることも有用であると考えられるが，それらを対応付けるためにはまず段落の粒度での対応関係を把握することが有効であると考えた．提案手法は 3 ステップから成り，各ステップでの処理の内容はそれぞれ 3.1, 3.2, 3.3 節にて説明する．

3.1 レーベンシュタイン距離による対応付け

最初のステップではレーベンシュタイン距離 [3] [5] を応用することで対応付けを行う．レーベンシュタイン距離は入力された二つの文字列がどの程度異なるかを示すものであり，文字の削除，挿入，置換という編集操作を入力文字列同士が同じ文字列になるまで繰り返すことで計算する．その際，各編集操作にはコストが設定されており，最終的に必要とする編集操作のコストの和が最小となるように上手く操作を組み合わせる．長さがそれぞれ i, j である文字列 s と t の間のレーベンシュタイン距離 $d(s_i, t_j)$ は式 (1) ～ (4) のように再帰的に求めることができる．

$$d(s_0, t_0) = 0 \tag{1}$$

$$d(s_i, t_0) = d(s_{i-1}, t_0) + cost_{del}(s[i]) \tag{2}$$

$$d(s_0, t_j) = d(s_0, t_{j-1}) + cost_{ins}(t[j]) \tag{3}$$

```
a p p l e
| | | ＼
a p r i l
```

図 2　レーベンシュタイン距離を利用した対応付けの例

$$d(s_i, t_j) = \min \begin{pmatrix} d(s_{i-1}, t_j) & + & cost_{del}(s[i]), \\ d(s_i, t_{j-1}) & + & cost_{ins}(t[j]), \\ d(s_{i-1}, t_{j-1}) & + & cost_{ren}(s[i], t[j]) \end{pmatrix} \tag{4}$$

式中の s_i は s の先頭から i 番目までの部分文字列，$s[i]$ は s の i 番目の文字であり，t_j, $t[j]$ についても同様である．また，$cost_{del}$, $cost_{ins}$, $cost_{ren}$ はそれぞれ削除，挿入，置換の編集操作のコスト関数であり．それぞれ一つないしは二つの文字を引数に取る．

レーベンシュタイン距離は距離を求めるための尺度であるが，これを対応付けに応用することを考える．具体的には，式 (4) において，置換操作が選ばれた場合，つまり最下段の式の値が最小となる場合に $s[i]$ と $t[j]$ を対応付けるのである．例えば文字列 “apple” と “april” に対してレーベンシュタイン距離を求めるアルゴリズムを適用すると，結果は図 2 のようになり，文字列の文字単位での対応付けがなされていることが分かる．

本研究は文書間の段落単位での対応付けが目的であるので，文書を段落の列として捉え，これにレーベンシュタイン距離のアルゴリズムを適用し，対応付けを行う．対応付けのアルゴリズムを Algorithm 1 に示す．動的計画法を利用したレーベンシュタイン距離を計算するアルゴリズムを拡張したものであり，動的計画法で利用する表には「その時点での距離」と「その距離を計算するに至った対応付け」の組を格納する．例えば，$\{2, \{\{p_1, \Lambda\}, \{p_2, q_1\}, \{\Lambda, q_2\}\}\}$ という値が格納されていたとすると，これは「p_1 が削除，p_2 と q_1 が対応付けられ，q_2 が挿入された．そしてそれらのコストの和は 2 である」ということを表している．そしてアルゴリズムは最後には対応付けの結果を出力する．

また，レーベンシュタイン距離は各操作のコスト関数を工夫することによって自由に拡張することができる．ここでは，段落を扱うのに沿った形にコスト関数を定義する．類似度が高いもの同士を対応付けたいので，段落同士の類似度が高ければ高いほど置換操作のコストが低くなるように設定する．本研究では式 (5), (6) をコスト関数として用いる．

$$cost_{del}(p) = cost_{ins}(p) = \alpha \qquad (0 \leqq \alpha \leqq 1) \tag{5}$$

$$cost_{ren}(p, q) = 1 - similarity(p, q) \tag{6}$$

各コスト関数へは一つないしは二つの段落が引数として渡される．それぞれの意味合いは文字列を扱った場合と同様である．削除と挿入は対称な操作であるためコストは等しくなるように設定する．α は対応付けに関する類似度の閾値のような意味を持つパラメータである．$similarity$ は段落 p, q の類似度を 0 から 1 までの範囲で返す関数である．つまり，類似度が高ければ高いほど $cost_{ren}(p, q)$ は小さくなり，対応付けられやすくなる．このように，レーベンシュタイン距離のアルゴリズムに段落の類似度を利用したコスト関数を設定することで，段落の類似度と段落の順序関係を考慮した対応付けを行うことができる．

Algorithm 1 レーベンシュタイン距離を利用した対応付け

Input: $Doc_1 = \{p_1, p_2, ..., p_m\}, Doc_2 = \{q_1, q_2, ..., q_n\}$
Output: The result of matching.

```
D ← matrix(m + 1, n + 1)
D[0, 0] ← {0, {}}
for i ← 1 to m do
  /* append(list, item): Append item to the end of list. */
  D[i, 0] ← {D[i − 1, 0][0] + cost_del(p_i),
             append(D[i − 1, 0][1], {p_i, Λ})}
end for
for j ← 1 to n do
  D[0, j] ← {D[0, j − 1][0] + cost_ins(q_j),
             append(D[0, j − 1][1], {Λ, q_j})}
end for
for i ← 1 to m do
  for j ← 1 to n do
    d_1 ← {D[i − 1, j][0] + cost_del(p_i),
           append(D[i − 1, j][1], {p_i, Λ})}        /* delete */
    d_2 ← {D[i, j − 1][0] + cost_ins(q_j),
           append(D[i, j − 1][1], {Λ, q_j})}        /* insert */
    d_3 ← {D[i − 1, j − 1][0] + cost_ren(p_i, q_j),
           append(D[i − 1, j − 1][1], {p_i, q_j})}  /* rename */
    D[i, j] ← min_{for element[0]} (d_1, d_2, d_3)
  end for
end for
return D[m, n][1]          /* {{p_1, q_1}, {p_2, Λ}, {Λ, q_2}, ...} */
```

$similarity$ としては TF-IDF を用いて段落の特徴語ベクトルを生成し，それらのコサイン類似度を計算するというのが一つの単純な方法である．しかし，1 節で述べたように，段落の内容が大きく書き変わっていた場合には段落の内容の情報だけに頼ったこの方法では適切な対応付けは難しい．

そこで，対応する段落は対応する節内にそれぞれ位置している可能性が高いということに注目して，節の類似度と段落の類似度を組み合わせたものを $similarity$ として利用することを考える．段落の属する節 (あるいは小節) 同士の類似度を sim_{global}, 段落同士の類似度を sim_{local} とする (図 3)．sim_{global} は注目している段落の属する節間の類似度を TF-IDF, コサイン類似度を用いて計ったものであり，sim_{local} は注目している段落間の類似度を同様に TF-IDF, コサイン類似度で計ったものである．これらを用いると，式 (6) 中の $similarity$ は sim_{global} と sim_{local} の重み付き和として

$$similarity(p, q) = C sim_{global}(p, q) + (1 - C) sim_{local}(p, q) \tag{7}$$

と表される．ここで C は定数である．これにより，たとえある段落が大きく書き変わっていて段落間の類似度が小さい場合でも，その段落が属する節の類似度が目的の段落を対応付けるための助けとなることが期待できる．

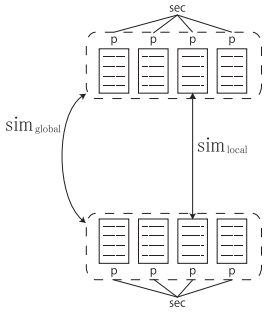

図 3　sim_{global} と sim_{local}

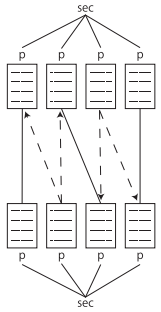

図 5　段落のマージ

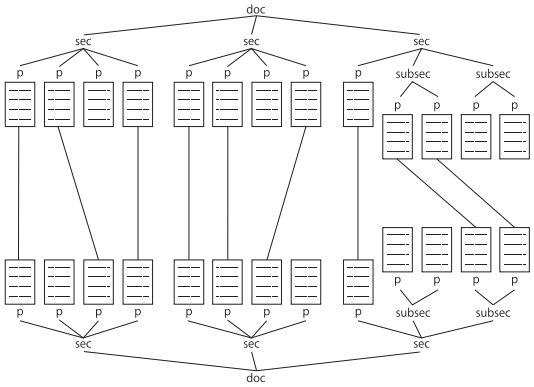

図 4　ステップ 1 での対応付けのイメージ図

ここまででレーベンシュタイン距離のアルゴリズムに基づいた対応付けを行う用意ができた．このステップでの対応付けのイメージ図を図 4 に示す．図では，対応付けられた段落が実線で結ばれている．対応付けられていない段落はこの段階では削除，あるいは挿入されたと見なされている段落である．もしこのステップで全ての段落が対応付けられた，もしくはどの段落も対応付けられなかった場合には，ここで対応付けの処理は終了する．

3.2　マ　ー　ジ

対応する文書間では，段落の分割が起こっている場合がある．前ステップまでの対応は 1 対 1 であったが，この場合，分割前の段落と分割後の段落とでの 1 対 多 の対応付けが必要となってくる．我々は段落のマージを行うことにより，1 対 多 の対応付けを実現する．

マージ処理の対象となるのは図 4 においてまだ対応が取れていない段落とする．文書中のある対応付けられていない段落に注目し，その両脇の既にどこかに対応付けられている段落を，マージ先の候補とする．これは，前ステップまでで分割前の段落と分割後のどれか一つの段落とが既に対応付けられていることを前提としていることになる．それぞれの段落のマージ先の候補を示したものが図 5 である．図中で，破線の矢印の元がマージ対象の段落，矢印の先がマージ後に対応付く段落，その段落と既に対応付いている段落がマージ先の段落の候補である．これら矢印で表されているマージの可能性それぞれについて，マージ操作を試していく．

注目している段落をマージするかどうかの判断は，マージを行った後の方が段落間の類似度が高くなるかどうかを調べることで行う．つまり，マージ後に対応付く段落から見て，既に対応付いている段落 (マージ対象の段落にとってのマージ先の段落) との類似度よりも，マージ対象の段落とマージ先の段落を結合した段落との類似度のほうが大きくなる場合にマージを行う．マージを行うことによって対応付けられた段落間の類似度が逆にマージ前よりも小さくなる場合には，そのマージは行われず，マージ対象となっていた段落は対応付けられないままとなる．

ここで，段落のマージについても，文書の構造を利用することを考える．段落のマージは段落の分割を想定した操作であるため，分割された段落が異なる節に分散することはないと考えられる．そこで，節をまたいだマージは起こりにくいと仮定し，マージが節をまたぐと判断された場合には，そのマージに関連する類似度を小さくする工夫を加える．ここでは，節をまたいだマージに対するペナルティを表す定数 $penalty$ を導入し，マージが行われる場合の類似度に掛け合わせる．マージが節をまたぐかどうかは，マージ対象の段落とマージ先の段落の親が同一であるかどうかで判断することができる．

ここまででマージ処理を踏まえた対応付けを行う準備ができた．前ステップで得られた対応付けの結果に，このステップで説明したマージ処理を行う．このステップでの対応付けのイメージ図を図 6 に示す．図中で，一つの段落に二つの段落が対応付けられている箇所がマージが行われた部分である．この例では 2 箇所でマージが実施されていることが分かる．また，マージ処理後でもまだ対応付いていない段落があることも分かる．これは，両脇の段落とのマージを試行してみたものの，元々対応付いていた段落間の類似度よりもマージした後の類似度の方が小さくなるためにマージが行われなかった部分である．このステップで全ての段落が対応付けられた場合には，対応付けの処理はここで終了する．

3.3　スワップ

ここまでのステップで複数段落間の対応付けがなされた．しかし，対応する文書間では，段落の順序が入れ替わっている場合がある．この場合，交差する対応付けを考慮する必要がある．このステップでは，交差する対応付け，即ち段落のスワップ処理を考える．

マージ処理と同じく，スワップ処理も処理の対象となるのは図 6 においてまだ対応付いていない段落である．前ステップまでの対応付けでは，順序の入れ替わりが起こった段落は対応な

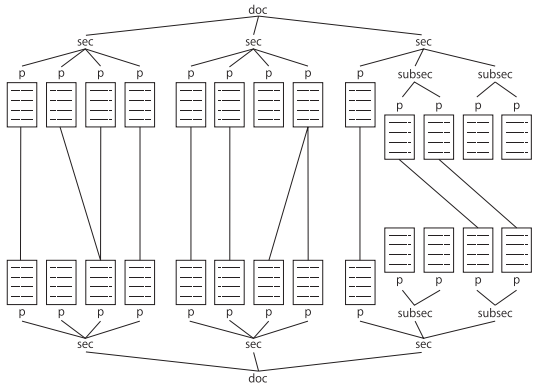

図 6　ステップ 2 での対応付けのイメージ図

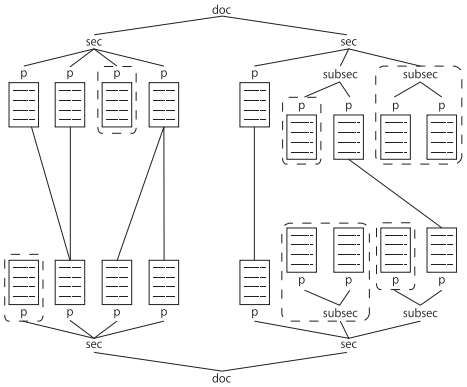

図 7　スワップの対象となる部分木の例

しのまま残っている可能性が高い．スワップ処理では，未だ対応付いていない段落に注目し，他方の文書中の同じように対応付いていない段落の中から対応するものを探す処理を行う．

スワップ先の段落を探す手掛かりはやはり段落間の内容の類似度を見ることであり，類似度が閾値を超えた場合に対応付けを行うことが考えられる．しかし，前ステップまでと同じように，対応する段落が書き変わっている場合には適切なスワップ先を選択するのは難しい．また，これまでとは異なり，スワップ処理は元々の文書の段落の順序を無視する自由度の高い処理であるので，誤った対応付けがなされてしまう可能性が高くなる．そのため，スワップを行うかどうかを判断する類似度の閾値は比較的高めに設定する必要がある．つまり，注目している段落間の類似度が十分に高くなければスワップ処理は行われないということである．

そこで，スワップ処理でも文書の構造を利用することを考える．対応付いていない段落単体をスワップ対象とするのでなく，文書木の中で対応付いていない段落のみを葉ノードとして持つ部分木を考え，注目している段落がある部分木に含まれていればその部分木を代わりにスワップ対象とするのである．図 7 にスワップ対象となる部分木の例を示す．図中で波線で囲まれている部分がスワップ対象となる．そして，スワップ対象となった部分木と，他方の文書中のスワップ先候補である部分木との類似度が閾値以上ならスワップ処理を行うようにする．これによって，段落単体の内容の類似度だけではスワップが行われなかった段落でも，同じ部分木に属する他の段落の類似度や部分木の構造的な情報の助けを借りて適切にスワップを行うことが出来るようになる．

部分木間の類似度の計算には Tree Edit Distance (TED) [11] を利用する．ここで TED を用いるのは，段落の内容の情報だけでなく，部分木の構造の情報も類似度として取り入れることによって内容の類似度が小さくても正しいスワップ先を見つけられることが期待できるためである．スワップ先候補の部分木の中で，閾値以上の類似度を持ち，かつ最も類似度の高いものをスワップ先として選択する．

スワップ先が決まったら，それら部分木に対して再度ステップ 1 から処理を行っていく．対応付けの処理はこのように再帰的に実行される．再帰が止まるのは，次のいずれかの場合である．

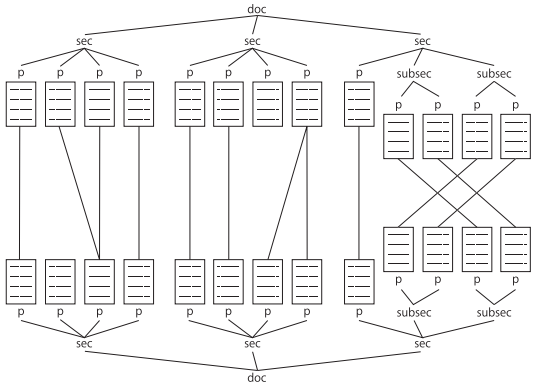

図 8　ステップ 3 での対応付けのイメージ図

（1） ステップ 1 で全ての段落が対応付けられた，もしくはどの段落も対応付けられなかった場合

（2） ステップ 2 で全ての段落が対応付けられた場合

（3） ステップ 3 でどの段落もスワップ先が見つけられなかった場合

このステップでの対応付けのイメージ図を図 8 に示す．図中で，対応付けが交差している箇所がスワップが行われた部分である．また，ここでも対応付いていない段落は，スワップ先が見つけられなかったか，スワップ先を見つけられたもののやはり対応付けられなかった部分である．

4. 実　　験

4.1 実験の準備

前節で提案した手法を評価するために，実際に対応文書を用いて評価実験を行った．今回は対応文書として異なる会議で発表された論文の組を用い，事前に人手で行った対応付けを正解データと見なし，提案手法による対応付けの結果と比較した．なお，実験では式 (5) の α は 0.425，式 (7) の C は 0.2，マージが節をまたぐ場合のペナルティ $penalty$ は 0.8 に設定した．

実験に用いた論文は所属する研究室において過去に発表されたものを利用した．その際，入手した PDF フォーマットの論文を提案手法で処理するために事前に文書木を表現する独自の XML フォーマットに変換して使用した．

実験に用いた対応文書は表 1 の通りである．これらのうち，

日本語で書かれた論文に関しては自動翻訳を用いてあらかじめ英語に翻訳したものを使用した．

提案手法を評価するために，他にいくつかの比較手法を用意した．

- 単純内容比較法

段落間の順序関係を考慮せずに対応付ける手法である．提案手法では，ステップ 1: レーベンシュタイン距離による対応付け，ステップ 2: マージ，ステップ 3: スワップという 3 ステップで対応付けを行っている．しかし，マージ，スワップはそれぞれレーベンシュタイン距離による対応付けが (1) 1 対 1 による対応付けである．(2) 要素の入れ替わりに対応できない．という性質を持つために必要な処理である．ステップ 1 でレーベンシュタイン距離のアルゴリズムを利用したのは「元々の段落の順序関係を重視する」ためであり，またその方が適切な対応付けが行われると考えたためである．

これを確認するために，単純内容比較法では入力を $Doc_1 = \{p_1, p_2, ..., p_m\}, Doc_2 = \{q_1, q_2, ..., q_n\}$ として，「$p_1, p_2, ..., p_m$ のそれぞれについて，$\max_{1 \leqq j \leqq n} (similarity(p_i, q_j))$ を与える q_j との類似度が閾値を超える時，即ち $1 - similarity(p_i, q_j) < 2\alpha$ ならば p_i と q_j を対応付ける」という処理を行う．ただし，ある段落が多数の段落と対応付くことを許容するとする．これにより，段落の順序を考慮しない対応付けを行う．

- 非構造利用法

本研究の主張は「文書の構造を利用することで対応付けが改善される」ということである．これを確認するために，構造を利用しない手法として非構造利用法を用意する．非構造利用法は，提案手法をベースに，以下の三つの修正を加えたものである．

（1） ステップ 1 (レーベンシュタイン距離による対応付け) において sim_{global} を用いない．つまり式 (7) の C を 0 に設定する

（2） ステップ 2 (マージ) において節をまたいだマージが行われる場合にも類似度に修正を加えない．つまり $penalty$ を 1 に設定する

（3） ステップ 3 (スワップ) においてスワップの処理単位は段落とし，部分木をスワップ対象としない

これにより，対応付けに利用できる情報は段落の内容と段落の順序関係だけとなる．

4.2 実験結果

用意した手法のそれぞれを対応文書に適用した結果を表 2 に示す．表内の数字の横に括弧書きで添えてあるのはその適合率，ないしは再現率の元となった分数であり，それぞれ

- 得られた対応付けのうち正解だったものの数 / 得られた対応付けの数
- 得られた対応付けのうち正解だったものの数 / 正解の対応付けの数

を表している．また，図 9 は提案手法，非構造利用法，単純内容比較法のそれぞれについて文書ペアごとの F 値を示したものである．

4.3 考　　察

図 9 を見ると，どの文書ペアについても F 値は 提案手法 ≧ 非構造利用法 ≧ 単純内容比較法 となっており，全体として提案手法が最も良い結果を示している．

単純内容比較法に注目すると，全体として他の二つの手法よりも対応付けの結果は劣っていることが分かる．結果を細かく見ると，単純内容比較法では段落の順序関係を無視した不自然な対応付けが多くなされていることが分かった．これに対して，他の二つの手法では大筋は元々の段落順序を再現した対応付け結果となっており，この差が結果の優劣に繋がったと考えられる．これにより，対応付けに段落の順序関係を考慮することは有効であると言える．

次に，提案手法と非構造利用法の結果を比較する．図 9 より，提案手法の方が，全体として非構造利用法よりも良い結果を示していることが分かる．対応付けの結果を見ると，提案手法において構造を利用した効果が各所で表れていた．また，文書ペアの種類に注目すると，言語が違うペア，特に 青戸 1，青戸 2，田邊，長谷川 で大きく改善が見られている．これは，処理前に自動翻訳にかけているため，内容の類似度では適切な対応が取れず，構造を利用することが有効な例が多くなったためであると考えられる．

提案手法において，文書の構造を利用したことで対応付けが改善された例を示す．

- ステップ 1 での改善例

ステップ 1「レーベンシュタイン距離による対応付け」では段落の内容を見るだけでは対応付けられない場合に対処するために，段落間の類似度 sim_{local} に加えて節間の類似度 sim_{global} を導入した類似度を利用したが，これによって青戸 1, 青戸 2, 田邊，長谷川，川本 で対応付けの改善が見られた．論文中には対応はしているものの内容が大きく書き変わっており，内容の類似度が低い段落が存在した．非構造利用法ではそれらの段落は対応付けられなかったが，提案手法のように節単位での類似度を考慮することにより，節内の他の類似度の高い段落の助けを借りて目的の段落を対応付けることができるようになった．

また，段落間の類似度だけを見ると一見対応付きそうでも実際は対応していない例もあった．非構造利用法では段落間の類似度だけで判断しているためそれらの段落を対応付けてしまった (誤った対応付け)．しかし提案手法では，それらの段落が属する節間の類似度が大きくないことを利用して，誤った対応付けを回避することができた．これも節間の類似度を利用することによる改善と言える．

- ステップ 2 での改善例

ステップ 2「マージ」では分割された段落が複数の節に分散することはないという仮定から，節をまたいだマージに対してペナルティを導入したが，これによって 田邊，長谷川，Zhang, Li で改善が見られた．論文中には元の段落が分割された訳ではないにもかかわらず，マージを行った方が段落間の類似度が大きくなる例があり，段落間の類似度だけでマージの判断を行う非構造利用法では，マージを行ってしまった (誤ったマージ)．しかし提案手法では，マージされる段落がそれぞれ異なる節に

表 1　実験に用いた文書

文書ペア名	用いた文書
青戸 1	・青戸 了，清水 敏之，増田 耕一，吉川正俊，“履歴情報管理システムのための多粒度アノテーション伝播”，DEIM 2010 ・R. Aoto, T. Shimizu, and M. Yoshikawa, “Propagation of Multi-granularity Annotations”, DEXA 2011
青戸 2	・青戸 了，清水 敏之，増田 耕一，吉川正俊，“履歴管理システムにおける多粒度アノテーション伝播”，DBSJ, Vol.9, No.1, 2010 ・R. Aoto, T. Shimizu, and M. Yoshikawa, “Propagation of Multi-granularity Annotations”, DEXA 2011
田邊	・田邊 翼，清水敏之，吉川正俊，“キーワードの役割と要素の特性を考慮した XML 検索”, WebDB Forum 2012 ・T. Tanabe, T. Shimizu, and M. Yoshikawa, “Effective Keyword-Based XML Retrieval Using the Data-Centric and Document-Centric Features”, AIRS 2012
長谷川	・長谷川 馨亮，馬 強，吉川 正俊，“行動の時空間連続性を考慮した旅行ツイートの組織化”, DEIM 2012 ・K. Hasegawa, Q. Ma, and M. Yoshikawa, “Trip Tweets Search by Considering Spatio-temporal Continuity of User Behavior”, DEXA 2012
川本	・川本 淳平，吉川正俊，“キー・バリュー型データベースにおける利用者のプライバシを考慮した範囲問合せの実現手法”, 情報処理学会論文誌データベース Vol. 4 No. 3, 2011 ・J. Kawamoto, M. Yoshikawa, “Private Range Query by Perturbation and Matrix Based Encryption”, ICDIM 2011
Takahashi	・A. Takahashi, M. Tatedoko, H. Kinutani, and M. Yoshikawa, “Metadata Management for Integration and Analysis of Earth Observation Data”, ICUIMC 2009 ・A. Takahashi, M. Tatedoko, T. Shimizu, H. Kinutani, and M. Yoshikawa, “Metadata Management for Integration and Analysis of Earth Observation Data”, Journal of Software, Vol.5, No.2, 2010
Zhang	・X. Zhang, Y. Asano, and M. Yoshikawa, “Analysis of Implicit Relations on Wikipedia: Measuring Strength through Mining Elucidatory Objects”, DASFAA, 2010 ・X. Zhang, Y. Asano, and M. Yoshikawa, “A Generalized Flow based Method for Analysis of Implicit Relationships on Wikipedia”, TKDE, Vol.25, No.2, 2013
Li	・J. Li, Q. Ma, Y. Asano, and M. Yoshikawa, “Ranking Content-Based Social Images Search Results with Social Tags”, AIRS 2011 ・J. Li, Q. Ma, Y. Asano, and M. Yoshikawa, “Improving Content-based Social Image Retrieval Based on an Image-tag Relationship Model”, IPSJ Transactions on Databases, Vol.5 No.3, 2012

表 2　評価実験の結果

	提案手法			非構造利用法			単純内容比較法		
	適合率	再現率	F 値	適合率	再現率	F 値	適合率	再現率	F 値
青戸 1	0.95 (36/38)	0.95 (36/38)	0.95	0.91 (32/35)	0.84 (32/38)	0.88	0.62 (31/50)	0.82 (31/38)	0.70
青戸 2	0.94 (34/36)	0.89 (34/38)	0.92	0.91 (30/34)	0.79 (30/38)	0.83	0.65 (24/37)	0.63 (24/38)	0.64
田邊	0.85 (34/40)	0.76 (34/45)	0.80	0.80 (33/41)	0.73 (33/45)	0.77	0.68 (34/50)	0.76 (34/45)	0.72
長谷川	0.86 (44/51)	0.92 (44/48)	0.89	0.81 (38/47)	0.79 (38/48)	0.80	0.44 (35/80)	0.73 (35/48)	0.55
川本	0.95 (37/39)	0.90 (37/41)	0.93	0.93 (37/40)	0.91 (37/41)	0.91	0.69 (31/45)	0.76 (31/41)	0.72
Takahashi	0.98 (51/52)	0.98 (51/52)	0.98	0.98 (51/52)	0.98 (51/52)	0.98	0.98 (51/52)	0.98 (51/52)	0.98
Zhang	1.0 (51/51)	0.93 (51/55)	0.96	0.98 (51/52)	0.93 (51/55)	0.95	0.88 (45/51)	0.82 (45/55)	0.85
Li	0.97 (37/38)	1.0 (37/37)	0.99	0.95 (37/39)	1.0 (37/37)	0.97	0.97 (31/32)	0.84 (31/37)	0.90

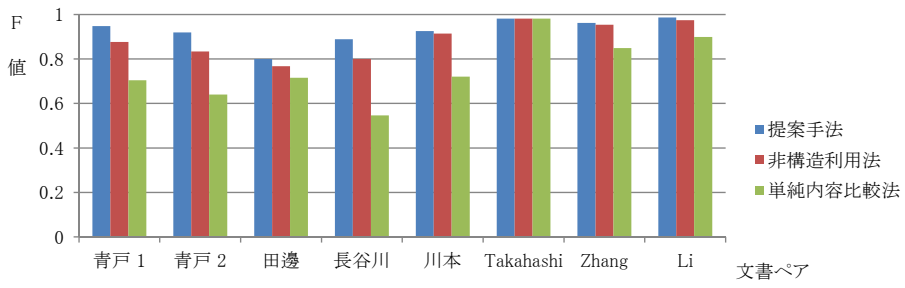

図 9　それぞれの手法での F 値の変化

属していることを検知して，誤ったマージを回避することができた．

- ステップ 3 での改善例

ステップ 3「スワップ」では段落単体の類似度だけではスワップ先を見つけられない場合に対処するために，スワップ対象として段落だけでなくそれを含む部分木も考慮することを考えた

が，これによって青戸 1, 青戸 2，長谷川 で改善が見られた．論文中には段落の順序が入れ替わっており，しかも内容が大きく書き変わっている例があった．非構造利用法ではスワップの対象は段落単位であり，段落間の類似度だけでスワップ先を見つけようとするため，類似度の低いそれらの段落はスワップ先を見つけられず，対応付けられなかった．しかし提案手法では，順序が入れ替わったのが段落単体ではなく節単位であることを利用し，節の構造や節内の他の段落の類似度も考慮して適切なスワップ先を見つけ，対応付けることができた．

このように，提案手法で導入した文書構造の利用が有効に作用し，対応付けの改善につながったことが分かる．以上より，文書間の細粒度での対応付けに文書の構造を利用することは有効であると言える．

次に，提案手法でも対応付けが改善しなかった例を示す．

対応している段落が対応付けられなかった例は 2 パターンあり，それぞれ

（1） ステップ 1 で，あまりにも類似度が低すぎて対応付けられなかった

（2） ステップ 2 で，マージした方が類似度が下がると判定されてマージされなかった

である．これらを対応付けるには，さらなる工夫が必要となる．

まず，ステップ 1 において sim_{global} を段落の属する節，あるいは小節同士の類似度としたが，ぞれぞれの段落について sim_{global} で注目する要素を固定せずに，比較する相手によって動的に要素を選択することが考えられる．つまり，比較する段落によって sim_{global} で注目する要素を節同士，小節同士，節と小節，小節と節などの選択肢の中から最適なものを選ぶようにするのである．これは，例えば節内をいくつかの小節に分割するような修正がなされている文書を扱う時に効果を発揮すると考えられる．これにより，段落間の類似度が極端に低い場合でも改善された節間の類似度を利用して対応付けられることが期待できる．

マージに関しても，提案手法では誤ったマージを避けるための工夫のみであったが，段落だけを見ていては難しいマージを支援するための工夫が必要であろう．同じ節内でのマージは実施されやすくする，ステップ 1 と同様に sim_{global} を導入する，などが考えられる．また，提案手法ではマージのペナルティを一定値としたが，これをマージ対象の段落間の文書木内での距離に依存したものにすることも考えられる．

5. おわりに

本研究では，文書間の細粒度での対応付けのために文書の構造を利用した手法を提案した．提案手法では段落の順序関係に配慮した対応付けをベースに，マージ，スワップを導入して要素の分割，順序の入れ替わりに対応した．評価実験では，構造を利用する提案手法が，その他の手法に比べて対応付けの適合率，再現率ともに高い値を示し，特に言語が異なる文書ペアを対応付ける場面で構造を利用することの有効性が確認できた．

今後の課題としては，第一により多くの対応文書の例を用いた評価実験を行うことが挙げられる．確かに実験では構造を利用した提案手法が良い性能を示したが，今回用いた例は提案手法を適切に評価するには少なすぎるため，「特定の例のみを正確に対応付けるための手法」であるという可能性もある．より多く，そしてより多様な対応文書の例を用いて評価実験を行う必要がある．また，実験を通して得られた知見を基に，より適切な対応付けが行えるよう提案手法を改良していくことも課題として挙げられる．

謝　辞

本研究の一部は科研費 22700097 の助成を受けたものである．

文　献

[1] Hal Daumé III and Daniel Marcu. A phrase-based hmm approach to document/abstract alignment. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pp. 119–126, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[2] Minwoo Jeong and Ivan Titov. Multi-document topic segmentation. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pp. 1119–1128, New York, NY, USA, 2010. ACM.

[3] VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, Vol. 10, p. 707, 1966.

[4] Joe Tekli and Richard Chbeir. A novel xml document structure comparison framework based-on sub-tree commonalities and label semantics. *Web Semant.*, Vol. 11, pp. 14–40, March 2012.

[5] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *J. ACM*, Vol. 21, No. 1, pp. 168–173, January 1974.

[6] Xiaojun Wan. A novel document similarity measure based on earth mover's distance. *Inf. Sci.*, Vol. 177, No. 18, pp. 3718–3730, September 2007.

[7] Sirvan Yahyaei, Marco Bonzanini, and Thomas Roelleke. Cross-lingual text fragment alignment using divergence from randomness. In *Proceedings of the 18th international conference on String processing and information retrieval*, SPIRE'11, pp. 14–25, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] Ching-Man Au Yeung, Kevin Duh, and Masaaki Nagata. Providing cross-lingual editing assistance to wikipedia editors. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part II*, CICLing'11, pp. 377–389, Berlin, Heidelberg, 2011. Springer-Verlag.

[9] Haijun Zhang and Tommy W.S. Chow. A coarse-to-fine framework to efficiently thwart plagiarism. *Pattern Recognition*, Vol. 44, No. 2, pp. 471 – 487, 2011.

[10] Haijun Zhang and Tommy W.S. Chow. A multi-level matching method with hybrid similarity for document retrieval. *Expert Systems with Applications*, Vol. 39, No. 3, pp. 2710 – 2719, 2012.

[11] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, Vol. 18, No. 6, pp. 1245–1262, December 1989.

[12] 丸川雄三, 岩山真, 奥村学, 新森昭宏. ローカルアラインメントを用いたテキスト間の柔軟な対応付け. 情報処理学会研究報告. 情報学基礎研究会報告, Vol. 2002, No. 87, pp. 23–28, sep 2002.

[13] 山本和英. 保険関連文書間の自動対応付け (産業日本語関連). *Japio year book*, Vol. 2011, pp. 274–279, 2011.