情 報 学 基 礎　68－4
自 然 言 語 処 理 151－4
（２００２．９．１７）

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

丸川　雄三† 岩山　真‡ 奥村　学† 新森　昭宏＊
maru@pi.titech.ac.jp iwayama@pi.titech.ac.jp oku@pi.titech.ac.jp shinmori@isl.intec.co.jp

†東京工業大学 精密工学研究所
‡東京工業大学 精密工学研究所 / 日立製作所
＊東京工業大学　総合理工学研究科　知能システム科学専攻
/ インテック・ウェブ・アンド・ゲノム・インフォマティクス

従来の DP マッチングでは難しかった交差の存在するテキスト間の対応付けを行う手法を提案する．提案手法の特徴は以下の二点である．まずはテキスト間における部分文字列同士のアラインメント，すなわちローカルアラインメントの概念と，その計算手法としてローカルアラインメント DP マッチングを導入した点であり，もう一点はローカルアラインメントの順位付けを行い，対応付けに利用した点である. 前者の工夫により, DP マッチングの利点である類似度の最適化と計算量の削減を実現し，後者の工夫により，交差にも対応したテキスト間の柔軟な対応付けを実現した．提案手法の適用例として，公開特許公報全文における「請求項」と「発明の詳細な説明」との対応付けを紹介し，本手法の有効性を議論する．

キーワード ： DP マッチング，ローカルアラインメント，特許

A Flexible Text Matching using Local Alignments

Yuzo MARUKAWA†, Makoto IWAYAMA‡, Manabu OKUMURA† and Akihiro SHINMORI＊
maru@pi.titech.ac.jp iwayama@pi.titech.ac.jp oku@pi.titech.ac.jp shinmori@isl.intec.co.jp

† Precision and Intelligence Laboratory, Tokyo Institute of Technology
‡ Precision and Intelligence Laboratory, Tokyo Institute of Technology and Hitachi, Ltd.
＊Department of Computational Intelligence and Systems Sciences, Tokyo Institute of Technology,
and
INTEC Web and Genome Informatics Corp.

A method of aligning a text with another text, in which the partial alignments include crossovers and overlaps, is proposed. This method has the following two characteristics. One is to introduce the concept of the local alignment between sub-strings and use the dynamic programming to enumerate the possible local alignments. Another is to extract sub-optimal local alignments in addition to the optimal one. The former realizes efficient enumeration of local alignments and the latter realizes flexible text matching, where the partial alignments have crossovers and overlaps. We show an example of applying the method for finding alignments between “claims” and “embodiments” in a patent application, and discuss its effectiveness.

Key words: Dynamic programming(DP) matching, Local alignment, Patent

1. はじめに

パラレルコーパスの普及に伴って，テキスト間の対応(アラインメント)を(半)自動的に同定する必要性が増してきた．多言語パラレルコーパスを例にとると，ある言語での言い回しと別の言語での言い回しとの対応が自動的に見つかれば，翻訳知識の学習に有用なデータを大量に集めることができる．パラレルコーパスとはいえ文書単位での粗い対応しかとれていない場合が多いため，上記のような細かい対応をとることは重要な課題として残っている．

単一言語内に限っても，要約と原文，予稿と講演書きおこし，など様々なパラレルコーパスがある．これらのパラレルコーパスでも，テキスト間の対応付けができれば，要約規則や言い換え規則などの学習が可能になる．

単一言語内でのパラレルコーパスでは，脱落挿入置換によってテキスト間の対応がとれる場合が多いため，DP マッチングを用いた対応付けが有効である．DP マッチングとは時系列データ間の最適な対応を動的計画法によって求める手法であり[1,7,9]，脱落挿入置換に強い．従来は音声認識[7]や遺伝子配列データの検索[1]で用いられていたが，最近ではパラレルコーパスの対応付けをはじめ，一般的なテキスト処理にも広く用いられている．

例えば，黒橋等[5]は DP マッチングにより日本語並列構造を検出している．山本等[10]は文書検索に DP マッチングを用いている．パラレルコーパスでの対応付けに関しては，加藤等[4]，村田等[6]の研究がある．これらは，比較的短いテキスト間の対応付けを対象にしていたが，対象テキストが長くなるにつれ DP マッチングにも限界が出てくる．

一つは交差の問題で，DP マッチングには交差に弱いという欠点がある．例えば「図書館で本を借りた」と「本を図書館で借りた」では，「図書館で」と「本を」の順序が交差しているため全体として対応をとることは難しい[1]．また，要約と原文の対応付けでは，原文に含まれる複数のテキスト断片がオリジナルの出現順序とは異なる順序で要約中に使われることも多く，このような場合も，要約と原文間でそのまま DP マッチングを適用することは難しい．そのため Jing 等[2]は，原文での単語出現位置を一度無視した上であらためて最適なアラインメントを決める手法を提案している[2]．

本論文では，複数のローカルアラインメント(部分対応)を同定することで上記の問題を解決する方法を提案する．複数のローカルアラインメントは，通常の DP マッチングを拡張することで効率良く求めることができる．この手法は遺伝子配列データの検索で用いられているが[1]，本論文では，テキスト断片間の対応付けにローカルアラインメントという考え方を導入することを提案する．

以下 2 節では，通常の DP マッチングを拡張して複数のローカルアラインメントを計算する方法を説明する．また，この手法をテキスト間への対応付けに適用する際の工夫や問題点を述べる．3 節では，特許文書内での対応付けに本手法を適用した例を紹介する．最後に 4 節で，まとめと今後の課題を述べる．

2. ローカルアラインメント

2.1 DP マッチング

文字列と文字列を比較して類似度を計算するためには，長さが異なる文字列同士を整列化して，両者の構成要素同士が比較できるようにする必要がある．各構成要素同士の類似度を総和することで文字列間の類似度が計算できる．整列化の方法としては，端点を一致させ，構成要素の出現順序を変更しないという制約を満たす「伸縮写像」を用いたものや，写像に逆写像を持たせることで対称性を保証し，さらに脱落を表す要素を導入した「脱落と挿入」などがある[9]．これらの整列化においては，構成要素の出現順序が保存されているという制約があるため，最大類似度の計算とそれを与える整列化パターンの決定アルゴリズムとして動的計画法を用いることができる．

例えば，整列化の方法として「脱落と挿入」を用い，「太郎が中学校に行った」と「次郎が高等学校に行った」の類似度を計算する．まず，準備として，文字列の構成要素を文字とし，文字要素間の類似度を次のように定義する．

一致	＋2
不一致	－2
読み飛ばし	－1

[1] 交差により文の意味が微妙に変わることもあるため，対応をとらないほうが良い状況もある．

[2] 最適なアラインメントを求める際に，原文での出現順序が改めて考慮される．

また，構成要素間の対応（整列化）は図１のような類似度テーブル内での経路で表現する．経路の種類は以下のとおりである．

←	文字 a_i を読み飛ばし
↑	文字 b_j を読み飛ばし
＼	a_i と b_j を対応させる

a_i は行方向文字列の先頭から数えて i 番目の文字要素を，b_j は列方向文字列の先頭から数えて j 番目の文字要素を表す．←は左の格子と経路を結び，↑は上の格子と，＼は左斜め上の格子と経路を結ぶことを意味する．これら以外の経路は認めない．

	＊	太	郎	は	中	学	校	へ	行	っ	た
＊	＼ 0	← -1	← -2	← -3	← -4	← -5	← -6	← -7	← -8	← -9	← -10
次	↑ -1	↑ -2	↑ -3	↑ -4	↑ -5	↑ -6	↑ -7	↑ -8	↑ -9	↑ -10	↑ -11
郎	↑ -2	↑ -3	＼ 0	← -1	← -2	← -3	← -4	← -5	← -6	← -7	← -8
は	↑ -3	↑ -4	↑ -1	＼ 0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
高	↑ -4	↑ -5	↑ -2	↑ -1	↑ -2	↑ -3	↑ -4	↑ -5	↑ -6	↑ -7	↑ -8
等	↑ -5	↑ -6	↑ -3	↑ -2	↑ -3	↑ -4	↑ -5	↑ -6	↑ -7	↑ -8	↑ -9
学	↑ -6	↑ -7	↑ -4	↑ -3	↑ -4	＼ -1	← -2	← -3	← -4	← -5	← -6
校	↑ -7	↑ -8	↑ -5	↑ -4	↑ -5	↑ -2	＼ 1	← 0	← -1	← -2	← -3
へ	↑ -8	↑ -9	↑ -6	↑ -5	↑ -6	↑ -3	↑ 0	＼ 3	← 2	← 1	← 0
行	↑ -9	↑ -10	↑ -7	↑ -6	↑ -7	↑ -4	↑ -1	↑ 2	＼ 5	← 4	← 3
っ	↑ -10	↑ -11	↑ -8	↑ -7	↑ -8	↑ -5	↑ -2	↑ 1	↑ 4	＼ 7	← 6
た	↑ -11	↑ -12	↑ -9	↑ -8	↑ -9	↑ -6	↑ -3	↑ 0	↑ 3	↑ 6	＼ 9

図 1：類似度テーブル

以上の条件下で，最大類似度を持つ経路を求める問題は，動的計画法によって効率良く解くことができる．図１の各格子内部の数字は上記条件を満たしながら動的計画法で求めた各格子までの最大スコアである．この図から，文字列間の最大類似度は右下の格子のスコア「9」となり，また，最大類似度を与えた経路は，右下の格子より経路を辿ることで求めることができる．最適経路を整列化パターンに変換すれば，「太郎は中学校へ行った」は，

太＊郎は中＊＊学校へ行った

となり，「次郎は高等学校へ行った」は，

＊次郎は＊高等学校へ行った

となる．ここで“＊”は読み飛ばしをあらわす．なお，各格子点では，最大類似度を与える経路を便宜上一意に決定しているが，もちろん複数経路を許してもよい．

2.2 ローカルアライメント

通常の DP マッチングでは，対応する要素が交差している場合に適切な対応付けが行なわれない．この問題を根本的に解決するためには，構成要素の出現順序が保存されているという制約を緩める必要があるが，長いテキスト間でマッチングを取る場合，このアプローチは計算量の増大を招く．

本論文では，複数の部分的な対応を見つけることで交差の問題を解決する手法を提案する．部分的な対応は DP マッチングを拡張することで容易に計算できる．この計算法は，ある機能を有する遺伝子配列に共通しているパターン（モチーフ）を見つけるためによく用いられている[1].

通常の DP マッチングでは，文字列同士の最大類似度とそのときの経路（アラインメント）が求まるが，それ以外の経路は捨てられてしまう．捨てられる経路の中には，部分的にアライメントが取れているものも存在する．この部分的なアライメントをローカルアラインメントと呼ぶ．例えば，「コンピュータを用いた数値計算手法」と「数値計算法とコンピュータの活用」では，「数値計算」と「コンピュータ」の文字列が一致し，アラインメントが取れる．しかし，両者の並びが交差しているため，通常のＤＰマッチングでは，両者が共に対応する整列化パターンは得られない．しかし，部分的に見ると，

コンピュータ＊＊を用
コンピュータの活＊用

や，

数値計算手法
数値計算＊法

の部分文字列における整列化パターンは，それぞれ高い類似度を持っている．そこで，高い類似度を持つ部分文字列間のローカルアラインメ

ントを複数個見つけることで，交差の問題が解決できるのではないかと考えた．

2.2.1 Local suffix alignment

複数のローカルアラインメントを求めるために，まず接尾部分文字列のアラインメント（local suffix alignment）を数えあげる．ここでのポイントは，接尾部分文字列として，空の文字列を許すことにある．空の文字列同士の類似度を０に設定することで，部分接尾文字列同士の類似度が０以下となった箇所では最適な接尾部分文字列が双方空となり，類似度も０にリセットされる．Local suffix alignment で最大値を持つものが最適なローカルアラインメントとなる．

Local suffix alignment それ自体は動的計画法を用いた通常の DP マッチングとほぼ同じ手順で求めることができる．異なるのは，類似度テーブル内で負となった格子点を０にリセットする点のみである．０となった格子点から新たな経路が始まることになる．このようにして作成した類似度テーブルを Local suffix alignment テーブルと呼ぶ．Local suffix alignment テーブルにおいて，最大値から順に格子点を選び，対応する経路を抽出すれば高い類似度を持つ部分文字列のペアを複数個見つけることができる．図２に，例文における local suffix alignment テーブルを示す. なお経路情報は省略してある．

2.2.2 ローカルアラインメントの探索

Local suffix alignment テーブルからローカルアラインメントを複数個選ぶ際に，単純に上位から格子点を選ぶと無意味な対応が数多く選ばれてしまう．なぜなら，１位の周辺経路のスコアはどうしても高くなるため，１位の部分文字列に読み飛ばしを加えただけのもの（枝）が高位の候補に入る可能性があるからである．そこで最後の文字いずれかが読み飛ばしである接尾部分文字列対は候補から除くことにする（枝打ち）．また，終点や始点が高位の経路における終点や始点と一致するものについても，高位の経路のバリエーションであるため候補から除くこととする．

その結果，図２においては，スコア「12」を付けた「コンピュータ」同士のアラインメントが１位となり，２位は，スコア「9」を付けた「数値計算＊法」と「数値計算手法」のアラインメントとなる．

	＊	コ	ン	ピ	ュ	ー	タ	を	用	い	た	数	値	計	算	手	法
＊	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
数	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0
値	0	0	0	0	0	0	0	0	0	0	0	1	4	3	2	1	0
計	0	0	0	0	0	0	0	0	0	0	0	0	3	6	5	4	3
算	0	0	0	0	0	0	0	0	0	0	0	0	2	5	8	7	6
法	0	0	0	0	0	0	0	0	0	0	0	0	1	4	7	6	9
と	0	0	0	0	0	0	0	0	0	0	0	0	0	3	6	5	8
コ	0	2	1	0	0	0	0	0	0	0	0	0	0	2	5	4	7
ン	0	1	4	3	2	1	0	0	0	0	0	0	0	1	4	3	6
ピ	0	0	3	6	5	4	3	2	1	0	0	0	0	0	3	2	5
ュ	0	0	2	5	8	7	6	5	4	3	2	1	0	0	2	1	4
ー	0	0	1	4	7	10	9	8	7	6	5	4	3	2	1	0	3
タ	0	0	0	3	6	9	12	11	10	9	8	7	6	5	4	3	2
の	0	0	0	2	5	8	11	10	9	8	7	6	5	4	3	2	1
活	0	0	0	1	4	7	10	9	8	7	6	5	4	3	2	1	0
用	0	0	0	0	3	6	9	8	11	10	9	8	7	6	5	4	3

図 2：Local suffix alignment テーブル

この例では，１位の経路まわりに，10-12 のスコアを持つ多数の「枝」が刈られている．また，スコア「11」を持つ「コンピュータ＊の活用」と「コンピュータを＊＊用」とのアラインメントは，１位の候補と始点が一致するため候補から除かれている．ここで，例えば読み飛ばしと不一致のペナルティーを低く設定すれば，「コンピュータ＊の活用」と「コンピュータを＊＊用」とのアラインメントが「コンピュータ」同士のアラインメントより上位になる可能性もあることに注意されたい．

このような「枝刈り」と，包含関係チェックにより，いたずらに多数の対応関係を取り出すことが抑制できる．

2.2.3 ローカルアラインメントを用いた対応付け

図３は，例文について，１位と２位のローカルアラインメントをテキスト中にタグ情報として挿入した例である．

<1>コンピュータ</1>を用いた<2>数値計算手法</2>
<2>数値計算法</2>と<1>コンピュータ</1>の活用

図 3：タグ付けの例

以上のように, DP マッチングの改良手法としてローカルアラインメントを用いることで交差が存在するテキスト間の対応付けが可能になる.

3. 特許公報への適用例

最初で述べたように，テキスト間の対応付けには様々な応用例がある．我々は，特許（明細書）を対象として,「請求項」と「発明の詳細な説明」との対応付けをとることを試みている[3].「請求項」は独特のスタイルで記述されるため[8], 非専門家には読みにくい．一方,「発明の詳細な説明」は比較的平易なスタイルで記述されている．また，請求項には不足している文字通り詳細な説明も含んでいる．よって,「請求項」の各部分に対応する「発明の詳細な説明」の部分が同定できれば，特許の読解支援として有用である.

本節では，提案手法の応用例として，本手法を特許文書に適用した例を述べる．対応付けを実施するテキストとしては，一方を「請求項」とし，もう一方を「発明の詳細な説明」の一部分とした.

また，文字を要素とし，要素同士の類似度および読み飛ばしスコアは,

一致	＋2
不一致	－2
読み飛ばし	－2

とした.

3.1 適用結果

適用例として以下の特許を取り上げる.

◎公開番号「特開平 11-321076」
◎発明の名称「インクジェット記録用粘着紙」

この特許の公開特許公報全文から,【請求項１】と,【発明の詳細な説明】の【０００９】～【００１３】を抜き出し，前節で提案したローカルアラインメントによる両者の対応付けを行った.結果を図４に示す.(a)が,【請求項】であり,(b)が【発明の詳細な説明】である．タグの番号は類似度順を０位から，また同じ番号のタグに挟まれた部分文字列は両者が対応していることを示している.

<6>インクジェット記録用</6>表面基材、粘着剤層、および剥離紙を積層してなる**<3>インクジェット記録用粘着紙</3>**において、**<0>該剥離紙が<8><5>フリーネス１３０～５５０ｍｌＣＳ</8>F の木材パルプ</5>を主原料と</0>**して抄紙し、**<4>基紙の両面をポリエチレン<7>でラミネート処理</7>し</4>**、少なくとも片面に、剥離剤層を設け、**<11>ラミネート処理</11>**前の該**<10><1>基紙の水分が６．０</10>～<9>１２．０重量％</9>で</1>**あることを特徴とする**<2>インクジェット記録用粘着紙</2>**。

(a)：請求項１

【０００９】本発明は**<0>該剥離紙のフリーネスが１３０～５５０ｍｌＣSF の木材パルプを主原料と</0>**するものである。**<5>フリーネスが１３０ｍｌＣSF 未満の木材パルプ</5>**は抄紙叩解処理の強化のためコストがかかりすぎ安価な剥離紙には使用困難である。また、**<8>フリーネスが５５０ｍｌCS</8>**Fを越えると緻密な紙層が形成できず好ましくない。
【００１０】本発明に用いられる剥離紙基紙はクラフト紙、クレーコート紙、グラシン紙、上質紙、模造紙等が上げられる。とりわけ、安価なクラフト紙、上質紙が好ましい。尚、米坪３０～３００ｇ／m2 程度の各種繊維シート類が用いられる。とりわけ７０～１５０ｇ／m2 程度のものが加工適性の面で好ましい。
【００１１】一方、本発明において必須の剥離紙**<4>基紙両面へのポリエチレンのラミネート処理は押し</4>**出し塗工機等の一般に知られている方法で処理される。ポリエチレンは低密度ポリエチレン、中密度ポリエチレン、および高密度ポリエチレン等が混合または単独で適宜使用される。また、ラミネート種類はマット、セミマット、ミラー、超ミラー、セミミラー等の処理方法を適宜使用してもよい。なお、ラミネート量は７～４０μm程度が好ましい。
【００１２】なお、本発明は**<1>基紙の水分を６．０～１２．０重量%<7>で</1><11>ラミネート処理</7></11>**することが重要である。**<10>基紙の水分６．０</10>**重量%未満であれば、粘着加工後の剥離紙水分がさらに低くなるため周囲の水分と極めて反応しやすくなり剥離紙自体がカールを発生しやすい。また、**<9>１２．０重量%</9>**を越えるとブリスターが発生しやすく好ましくない。
【００１３】剥離剤としては特に限定されるわけではなく各種のシリコーン化合物やフッ素化合物が常法に従って塗布される。なお、シリコーン剥離剤は通常トルエンやヘキサン等の有機溶剤に溶解して塗布される。しかし、この塗布液として、熱、紫外線あるいは電子線で硬化させる無溶剤方式においても本発明の**<6><3><2>インクジェット記録用</6>粘着紙</2></3>**は本発明所望の優れた性能を発揮する。

(b)：発明の詳細な説明

図 4：特許における対応付け例

3.2 結果の考察

まず，当初の目的である交差を含む文字列の対応付けを幾つか見ることができる．特に<1>と<11>の対応付けが，本手法の効果を確認する好例となっている．「ラミネート処理前の該基

紙の水分が６．０～１２．０質量%で」と「基紙の水分を６．０～１２．０質量%でラミネート処理」の対応がとれていることに注目してほしい．

また本手法は，ローカルアラインメントの特徴として対応の重複を許すため，片方に一回，もう片方に複数回出現する文字列についても対応付けがなされる．この例を<2>, <3>や, <7>, <11>について見ることができる．なお，<2>, <3>については，対応元，対応先とも同一文字列の同スコアであり，順位の違いに意味はない．

更に<0>，<5>，<8>では，請求項で「A～B の C を主原料として」と記述されている部分が，発明の詳細な説明では「A～B の C を主原料とするものである。A 未満の C は･･･B 以上は…」と詳細な説明が付記されている．本手法を用いると，このような対応も見つけることができる．<1>，<9>，<10>も同様な例である．なお，Jing 等の方法[2]では対応の重複を許さないため，上記のような対応を見つけることはできない．

最後に，個々の対応では全て，脱落挿入置換が考慮されていることに注目して欲しい．脱落挿入置換の典型的な例は<4>に見ることができる．ここでは「基紙の両面をポリエチレンでラミネート処理･･･」と「基紙両面へのポリエチレンのラミネート処理･･･」との対応がとれている．

4. おわりに

本論文では，ローカルアラインメントによりテキスト間の柔軟な対応付けを行う方法を述べた．ローカルアラインメントは通常の DP マッチングを拡張することで効率良く数えあげることができる．この方法は遺伝子配列の検索において既に提案されているが，本論文ではテキスト間の対応付けに適用することを提案し，交差を含むテキスト間の対応付けも行えることを示した．また，実際の例として，特許における「請求項」と「発明の詳細な説明」との対応付けを示した．

以下今後の課題について述べる．

1. 本手法の評価を行う必要がある．現在，特許を題材に評価用データを作成している．
2. 対象テキストが長くなると実行速度が低下するため，何らかの高速化を行う必要がある．
3. 文字単位ではなく単語単位で対応をとることも必要である．
4. ツールとして整備する．遺伝子配列の検索に関してはBLAST, FASTA等のアラインメント用プログラムがあるが，本ツールはテキストに特化し，かつ研究者が自由にカスタマイズできるものを目指す．

参考文献

[1] Gusfield, D., Algorithm on Strings, Trees, and Sequences, Cambridge University Press, 1997.

[2] Jing, H., McKeown, K. R., The Decomposition of Human-written Summary Sentences, Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.129-139, 1999.

[3] 岩山真, 藤井敦, 高野明彦, 神門典子, 特許コーパスを用いた検索タスクの提案, 情報処理学会情報学基礎研究会, 2001-FI-63, pp.49-56, 2001.

[4] 加藤直人, 浦谷則好, 局所的要約知識の自動獲得方法, 自然言語処理, Vol.6, No.7, pp.73-92, 1999.

[5] 黒橋禎夫, 長尾眞, 並列構造の検出に基づく長い日本語文の構文解析, 自然言語処理, Vol.1, No.1, pp.35-57, 1994.

[6] 村田真樹, 井佐原均, diff と言語処理, 情報処理学会自然言語処理研究会, 2001-NL-144, pp.127-134, 2001.

[7] 中川聖一, 確率モデルによる音声認識, 電子通信情報学会(コロナ社), 1988.

[8] 新森昭宏, 奥村学, 丸川雄三, 岩山真, 手がかり句を用いた特許請求項の修辞構造解析, 2002-NL-149, pp.65-72, 2002.

[9] 上坂吉則, 尾関和彦, パターン認識と学習のアルゴリズム, 文一総合出版, 1990.

[10] 山本英子, 梅村恭司, 小澤智裕, 山本幹雄, 一般化文字列類似度を用いた文字ベースの情報検索, IREX ワークショップ論文集, pp.95-100, 1999.