アラインメントに基づくミスマッチクラスタからの最小汎化集合の抽出

宮原 和也† 田村 慶一‡ 北上 始‡

†広島市立大学情報科学部 〒731-3194 広島県広島市安佐南区大塚東 3-4-1
‡広島市立大学大学院情報科学研究科 〒731-3194 広島県広島市安佐南区大塚東 3-4-1
E-mail: †kazuya@de.info.hiroshima-cu.ac.jp, ‡{ktamura,kitakami}@hiroshima-cu.ac.jp

あらまし 曖昧な問合せ処理では，非常に多くの類似した部分文字列(ミスマッチクラスタ)が検索結果として得られる．ミスマッチクラスタを一般のユーザが直接見て，規則性を発見するのは非常に困難である．そこで，ミスマッチクラスタから最小汎化集合と呼ばれる集合を抽出する手法が研究されている．しかしながら，従来の研究では，異なる文字列長の部分文字列から構成されるミスマッチクラスタから最小汎化集合の抽出ができないという問題点が存在する．そこで，本研究では，異なる文字列長の部分文字列で構成されたミスマッチクラスタに対してアラインメント処理を行い，アラインメント結果から最小汎化集合を抽出する手法を提案する．提案手法では，ミスマッチクラスタの部分文字列の長さをアラインメント処理により整え，長さを整えたミスマッチクラスタから汎化処理により最小汎化集合を抽出する．提案手法により，文字列長が異なる部分文字列から構成されたミスマッチクラスタから最小汎化集合を抽出することが可能となる．

キーワード テキストマイニング，データマイニング，汎化処理

The Alignment-based Extraction of Minimum Generalized Set from a Mismatch Cluster

Kazuya MIYAHARA† Keiichi TAMURA‡ and Hajime Kitakami‡

†Faculty of Information Sciences, Hiroshima-City University 3-4-1 Ozuka-Higashi, Asa-Minami-ku, Hiroshima, 731-3194 Japan
‡Graduate School of Information Sciences, Hiroshima-City University 3-4-1 Ozuka-Higashi, Asa-Minami-ku, Hiroshima, 731-3194 Japan
E-mail: †kazuya@de.info.hiroshima-cu.ac.jp, ‡{ktamura,kitakami}@hiroshima-cu.ac.jp

Abstract An ambiguous query processing returns a large number of similar subsequences, called a mismatch cluster, to the user. It is difficult for the user to discover the characteristics from a mismatch cluster. In order to support user, the method of extracting a reduced expression, called minimum generalized set, of the mismatch cluster have been proposed. However, the existing proposed method cannot extract the minimum generalized set from the mismatch cluster which consists of substrings of a different string length. This paper a novel method of extracting the minimum generalized set from the mismatch cluster which consists of substrings of a different string length. In the proposed method, first, the length of the substring of a mismatch cluster is first prepared by alignment processing. Next, the minimum generalization set is extracted from the mismatch cluster which prepared length by generalization processing. The proposed method can extract the minimum generalization set from the mismatch cluster which consists of substrings of a different string length.

Keyword Text Mining, Data Mining, Generalized Processing

1. はじめに

曖昧な問合せ処理は，テキストデータベースや配列データベースから類似する部分文字列の検索をさし，Web文書，オンライン文書，分子配列データなどに対する情報検索を初めとして，クラスタリングや配列データマイニングなどの多くの分野で重要な要素技術である．曖昧な問合せ処理は，人為的過誤による誤字を含む単語や熟語を初めとしてカタカナ語の異表記同義語などが含ま

れるテキストデータ，同じメッセージが通信路ノイズと共に繰り返し現れるストリームデータ，わずかなゆらぎをもつモチーフが含まれるアミノ酸の分子配列データなどに対する類似検索に有用である．

曖昧な問合せ処理では，非常に多くの類似した部分文字列が検索結果として得られる．本論文では，曖昧な問合せ処理の結果として得られる部分文字列の集合をミスマッチクラスタと呼ぶ．ユーザがミスマッチクラスタを直接閲覧し，規則的な特徴を把握することは極めて困難である．そこで，ミスマッチクラスタから最小汎化集合と呼ばれる，極大な汎化配列パターンの集合と汎化できなかった部分文字列を抽出する研究[1]～[4]が行われている．最小汎化集合とは，ミスマッチクラスタをカバーする最小の汎化集合をさす，すなわち，それ以上に汎化するとミスマッチクラスタの要素とは無関係の要素を含んでしまう汎化集合をさす．例えば，文字列の集合{<*AB*>,<*CD*>,<*AD*>,<*CB*>}は，正規表現の配列パターン<[*AC*][*BD*]>として表現することができる．本研究では，この配列パターンのことを，汎化配列パターンと呼ぶ．また，汎化配列パターンが異なる配列データに現れる数を汎化配列パターンの支持数と呼ぶ．

最小汎化集合を抽出することで，ユーザは，(1) 曖昧な問合せ処理の結果として得られる部分文字列のすべてを閲覧・分析する手間から解放され，(2) 部分文字列の規則的な変化の様子を理解できる．例えば，文献[2][3]では，アミノ酸配列データを用いた実験において，ミスマッチクラスタから最小汎化集合を抽出することで，汎化配列パターンを支持数でランキングすると，生物学的な機能をもつモチーフと呼ばれる配列パターンが，上位にランキングされる傾向があることが確認されている．

このように，ミスマッチクラスタから最小汎化集合を抽出することは，規則的な特徴を把握することに有効であるが，従来の手法[1]～[4]では，ミスマッチクラスタを構成する部分文字列の文字列長が等しい場合でしか最小汎化集合を抽出することができないという問題点が存在する．従来の手法は，部分文字列の同じ列を汎化の対象としており，同じ列に存在する文字を正規表現に変換することを前提に作成されている．

本研究では，従来手法を有効活用するという前提のもとにモチーフの抽出精度をさらに向上させるために，異なる文字列長の部分文字列で構成されたミスマッチクラスタに対してアラインメント処理を行い，アラインメント結果から最小汎化集合を抽出する手法を提案する．提案手法では，(1) ミスマッチクラスタ内の各部分文字列の長さをアラインメント処理によって整え，(2) 長さを整えたミスマッチクラスタから汎化処理により最小汎化集合を抽出する．本論文で提案する提案手法を使用することにより，従来扱うことができなった種類のミスマッチクラスタから最小汎化集合を抽出することが可能となる．

提案手法を実際に実装し，試験データを用いて評価実験を行った．評価実験で，試験データから最小汎化集合を抽出したところ，提案手法を用いることで，文字列長が異なる場合でも最小汎化集合を抽出することができた．

本論文の構成は以下の通りである．第 2 章では関連研究として汎化処理について述べる．第 3 章で問題の定義を示し，第 4 章で提案手法について説明する．第 5 章で評価実験結果を示し，第 6 章で本論文のまとめを行う．

2. 関連研究

曖昧な問合せ処理の研究は，ある文字列に対して，文字の削除，挿入，置換をする操作に基づく類似性検索や近似検索の研究として数多く行われてきた．これらの研究では，類似検索性能の向上が中心話題である．このため，大量に返される類似検索結果を分析し規則的な特徴を把握する方法には十分な関心が寄せられていない．本論文では，この点に着目し，ミスマッチクラスタから最小汎化集合を抽出する手法について提案する．

ミスマッチクラスタから最小汎化集合を抽出する効率的な方法として，反復精密化法[1][2]，ドメイン分割法と反復精密化法の併用方法[1][2]，段階的一般化法[3][4]が提案されている． 反復精密化法は，ミスマッチクラスタから最汎パターンと呼ばれるミスマッチクラスタを表現する最も一般的な汎化配列パターンを起点を探索木のルートとし，パターン切除に基づき，ルートから下方向に探索を進める．段階的一般化法は，ミスマッチクラスタを構成する部分文字列の部分集合に対す

る汎化配列パターンを段階的に列挙することで探索を進める．

反復精密化法や段階的一般化法を用いることで，ミスマッチクラスタから最小汎化集合を求めることができるが，これらの手法は，ミスマッチクラスタを構成するすべての部分文字列の長さが等しいことを前提に作成されている．本研究では，ミスマッチクラスタを構成する部分文字列の長さがそれぞれ異なることを前提としているため，従来の手法をそのまま使用することができない．

3. 記号と問題の定義

本章では，本論文で使用する記号と最小汎化集合抽出問題の定義を行う．

3.1. ミスマッチクラスタ

本論文では，ミスマッチクラスタ *MIS* を次の形式で表現する．

$$MIS=\{ <inst_1>, <inst_2>, \cdots, <inst_n> \} \quad (1)$$

例えば，部分文字列，“*ABC*”，“*ABDC*”，“*ACC*”と“*ABBC*”とから構成されるミスマッチクラスタは，*MIS* = {<*ABC*>, <*ABDC*>, <*ACC*>, <*ABBC*>}と表記される．

また，曖昧な問合せ処理によってデータベースから得られた部分文字列<$inst_i$>をインスタンスと呼ぶ．各インスタンス<$inst_k$>の長さを|<$inst_k$>|と表し，インスタンス<$inst_k$>の第 *i* 文字目の文字を<$inst_k$>[*i*]と表記する．例えば，*MIS* = {<*ABC*>, <*ABDC*>, <*ACC*>, <*ABBC*>}を考えると，<$inst_1$>[1]=“A”，< $inst_1$>[2]=“B”，< $inst_1$>[3]=“C”である．

3.2. 汎化配列パターン

記号 Σ_i を任意の 1 文字 Σの部分集合とするとき，次式のように *k* 個の Σ_i を並べたパターンを *k*-汎化配列パターンと呼び，<pat^k>と表記する．

$$<pat^k>=<\Sigma_1\Sigma_2\ldots\Sigma_{k-1}\Sigma_k> \quad (2)$$

ただし， Σ_i は，たびたび括弧[ ]の中に Σ_i の全要素を列挙した表記をする．$\Sigma_i \subseteq \Sigma$が存在する場所を曖昧文字領域と呼ぶ．また，$|\Sigma_i| \leqq 2$ のとき，集合 Σ_i は曖昧文字ドメインと呼ばれ， Σ_i 内に存在する任意の 1 文字の配置が許されることを示している．曖昧文字ドメインが 1 個所以上存在するとき，<pat^k>を *k*-汎化配列パターンと呼ぶ．

例えば，汎化配列パターン<[*AB*][*CD*]>について考える．汎化配列パターン中の[*AB*]と[*CD*]は，それぞれ， Σ_1 ={*A*,*B*}， Σ_2 ={*C*,*D*}をさし，1 文字目は，“*A*”または“*B*”が，2 文字目は，“*C*”または“*D*”であることを示している．

また，関数 *EVAL* は汎化配列パターンに含まれるすべてのインスタンスを求める関数とする．例えば，*EVAL*(<[*AB*][*CD*]>)={<*AC*>, <*AD*>, <*BC*>, <*BD*>}である．

ここで，式（2）は， Σ_i の 1 つの要素が文字列である場合も許すように式（2）を一般的な定義に直す．Σから有限個の文字を取り出し，左から右へ 1 列に並べた文字列を Σ-文字列と呼び．Σ-文字列の全体を Σ^*と表記する．ここで，記号 Σ^*_i を任意の文字列 Σ^*の部分集合とするとき，*k*-汎化配列パターンを以下のように再定義する．

$$<pat^k>=<\Sigma^*_1\Sigma^*_2\ldots\Sigma^*_{k-1}\Sigma^*_k> \quad (3)$$

ただし， Σ^*_i は，たびたび括弧[ ]の中に Σ^*_i の全要素を列挙した表記をする．ただし， Σ^*_i の要素は文字列であるため，文字列の区分を示すために文字列の間に記号“|”を入れて区別する．

3.3. 最小汎化集合の抽出

本研究のゴールは，ミスマッチクラスタ *MIS* を構成する部分文字列の長さが一定ではないときに，そのミスマッチクラスタ *MIS* から最小汎化集合 *MGS* を抽出することにある．

集合 *MGS* = {G_1,G_2,…,G_m}(1≦*m*≦|*MIS*|)が，*k*-汎化配列パターン<pat^k>および *k*-インスタンス<$inst^k$>から構成されているとする．ただし，*EVAL*(<pat^k>)⊆*MIS* かつ，<$inst^k$> ∈*MIS* を満たすものとする．この集合 *MGS* が以下の性質を満たすとき，*MGS* を *MIS* に対する最小汎化集合と呼ぶ．

(1) *EVAL*(*MGS*)= *MIS*

(2) *MGS* の任意の 2 つの要素 G_i，G_j に対して，G_i と G_j の間には冗長な関係が存在しない（$1 \leqq i \neq j \leqq m$）．

(3) *MGS* に含まれるどの要素 G_i も極大（さらに汎化すると *MIS* に存在しないインスタンスを含んでしま

う）である．

(4)　上記の（1）～（3）を満たす任意の *MGS'* に対して，$|MGS'| \leqq |MGS|$が成立する．

一般に，*MGS* が上記（1）～（3）を満たすだけでは *MGS* を一意に定めることができないので，これらに上記（4）を追加し，最小汎化集合が一意に定まるようにしている．例えば，*MIS*={<*ABC*>,<*ABD*>,<*ACD*>,<*BCD*>}とすると，MGS_1 = {<*AC*[*CD*]>,<[*AB*]*CD*>,<*A*[*BC*]*D*>}と MGS_2 = {<*AB*[*CD*]>, <[*AB*]*CD*>}の 2 つの汎化集合はどちらも（1）～（4）を満たすが，（4）を追加することにより，MGS_1 が最小汎化集合として一意に選択される．

4. 提案手法

本章では，提案手法について説明する．4.1 節では従来の手法の問題点を述べ，4.2 節で全体の処理手順について説明を行う．次に，4.3 節で累進法によるマルチプルアラインメントについて示し，4.4 節ではアラインメント結果に対して汎化処理を行うことを可能とするための$変換について説明する．

4.1. 問題点

単純に，文字列長を整えるだけならば，ミスマッチクラスタのインスタンスに空白文字を挿入することで問題を解決することができる．例えば，*MIS'* = {<*ABC*□>, <*ABDC*>, <*ACC*□>, <*ABBC*>}として，空白文字を入れたとする（□は空白とする）．文字列長が同じになるので，汎化処理を行うことが可能となる．

しかしながら，汎化処理ではインスタンスの同じ列を 1 つの曖昧文字ドメインとして汎化するため，空白文字を文字列の最右端に挿入するのでは，意味のある最小汎化集合を求めることができない．*MIS* = {<*ABC*>, <*ABDC*>, <*ACC*>, <*ABBC*>}は，最初の文字が“*A*”で最後の文字が“*C*”が現れ，その間に“*B*”，“*BD*”，“*C*”，“*BB*”という文字列が現れるという規則性があると捉え，最小汎化集合として，*MGS*={<*A*[*B*|*BD*|*C*|*BB*]C>}が取り出せた方が最も自然である．しかしながら，空白文字を入れるのでは，最小汎化集合は，*MGS*= {<*ABC*□>, <*ABDC*>, <*ACC*□>, <*ABBC*>}となり，意味のある規則性を取り出すことができない．

そこで，ミスマッチクラスタのインスタンスについて類似性が高い文字が同じ列に配置されるようにアラインメント処理を行うことで，上述のような意味のある規則性を抽出できるようにする．アラインメントとは，文字列長が異なる文字列を類似する文字がなるべく多く同じ列に配置されるように，文字列に対して挿入・削除を意味するギャップ記号“－”を挿入することで文字列長を整えることである．

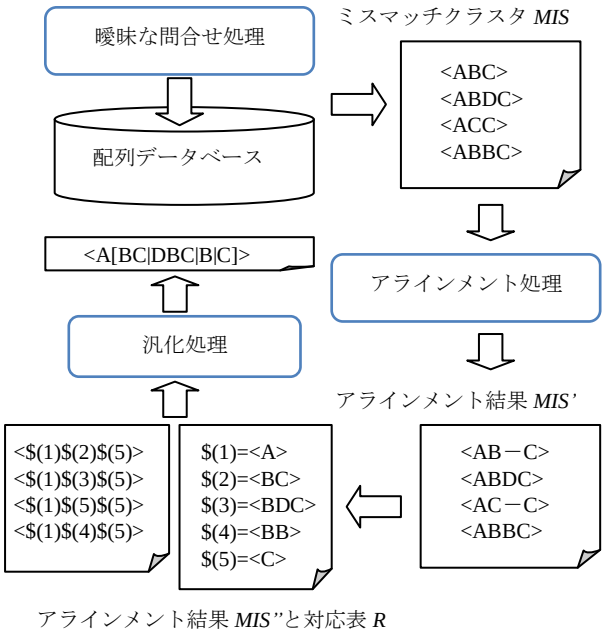

図 1 提案手法による処理手順の例

4.2. 処理手順

提案手法の処理手順を以下に示す．

（1）　配列データベース *D* に対して，キーワード *K*，誤差数 ε をそれぞれ入力し，曖昧な問合せ処理を行う．問合せ処理で得られた検索結果をミスマッチクラスタ *MIS* とする．

（2）　ミスマッチクラスタ *MIS* に対して，アラインメント処理を行い，ギャップ記号“－”を入れることでミスマッチクラスタの各インスタンス <$inst_k$>（$1 \leqq k \leqq n$）の長さを同じにする．本研究ではアラインメント処理には，マルチプルアラインメントのアルゴリズムとして提案されている累進法[6]を使用する．累進法によるマルチプルアラインメントに関しては 4.3 節で詳しく説明する．ここで，アラインメント処理を行った

後のミスマッチクラスタを *MIS'* とする．

(3) ミスマッチクラスタ *MIS'* に対して$変換を行う．$変換関数を *Dollar*(*X*)（*X* はミスマッチクラスタ）とすると *MIS"=Dollar*(*MIS'*)となる．$変換では，ミスマッチクラスタを構成するインスタンスを列ごとに比較し，列について，まとまりのある部分文字列を$(*n*)と表記する記号に変換する．また，変換元の文字列と変換後の$(*n*)記号の対応表 *R* を作成する．$変換については，詳しくは，4.4 節で説明する．

(4) $変換によって各インスタンスが$(*n*)の記号列に変換されたミスマッチクラスタ *MIS"* に対して汎化処理を行い，最小汎化集合 *MGS* を抽出する．汎化処理についての詳しい処理手順は，文献[1]～[4]を参照されたい．

(5) 対応表 *R* を使用して，$(*n*)の記号列を文字列に戻す．

図 1 に，提案手法による処理手順の例を示す．図の例は、*MIS*= {<*ABC*>, <*ABDC*>, < *ACC*>, <*ABBC*>}の提案手法による汎化処理を示している．

4.3. 累進法によるマルチプルアラインメント

累進法では，階層的に部分文字列をクラスタリングしながら，アラインメントを行っていく．以下に，累進法を用いたマルチプルアラインメントの処理手順を簡単に示す．

(1) ミスマッチクラスタ *MIS* に対して，インスタンスの全ての組み合わせ（<$inst_k$>,<$inst_l$>）について，ペアワイズアラインメントによりインスタンス間の最適アラインメントスコア $Score_{k,l}$ を求める．

$$Score_{k,l} = D_{|<inst_k>|,|<inst_l>|}$$

$$D_{i,j} = \max \begin{cases} D_{i-1,j-1} + GS(<inst_k>[i], <inst_k>[j]) \\ D_{i,j-1} - 1 \\ D_{i-1,j} - 1 \end{cases} \quad (4)$$

$$GS(x,y) = \begin{cases} x = y \text{のとき}, 1 \\ x \neq y \text{のとき}, -1 \\ x, y = \text{'–'} \quad \text{のとき}, -1 \end{cases} \quad (5)$$

(2) インスタンス間のすべての最適アラインメントスコア $Score_{k,l}$ を用いて類似度行列 *S* を作成する．

(3) 類似度行列 *S* を基に，階層的クラスタリングにより，案内木と呼ばれる木構造を作成する．

(4) 案内木を用いて，類似度が高い頂点から低い頂点へという順番で，インスタンス対インスタンス，インスタンス対クラスタ，クラスタ対クラスタのアラインメントを行う．案内木の根に相当する部分のアラインメント結果が最終的なアラインメントに対応する．

例えば, 図 1 の例では *MIS* に対してアラインメント処理をすると，*MIS'*= {<*AB－C*>, <*ABDC*>, < *AC －C*>, <*ABBC*>}が得られる．

4.4. $変換

アラインメント処理において，ギャップ記号“－”を入れることでミスマッチクラスタの各インスタンス<$inst_k$>（$1 \leqq k \leqq n$）の長さが同じになる．ただし，ギャップ記号は長さを整えるために便宜上入れた記号であるため，ギャップ付きのミスマッチクラスタをそのまま汎化することはできない．

そこで，ギャップ記号“－”を取り除いたとしても文字列長が整うように文字列にまとめ，まとめた文字列を記号$(*n*)に変換する．$(*n*)の *n* はまとめた文字列を順に 1 から番号付した整数値となる．汎化処理では，$(*n*)を 1 つの列単位として汎化処理を行う．

例えば，*MIS'*= {<*AB － C*>, <*ABDC*>, < *AC －C*>, <*ABBC*>}の$変換の例を示す．最初の 1 文字目は同じであるが，次の 2 文字目は次の文字は，文字目と異なる．そこで，$(1)=<*A*>とする．次の 3 文字目は 2 文字目と異なるが，ギャップ記号“－”があるため 2 文字目と結合する．ここで, $(2)=<*B*>, $(3)=<*BD*>, $(4)=<*BB*>, $(5)=<*C*>となる．4 文字目はギャップ記号“－”を含まず，すべて同じ文字“*C*”であり，$(5)=<*C*>がすでに存在するので，$(5)とする．

以上の変換により，*MIS"*={<(1)(2)$(5)>, <$(1)(3)(5)>, <(1)(5)$(5)>, <$(1)(5)(5)>}となる．$変換によって，ギャップ記号“－”を取り除いたとしても長さが 3 と整う． *MIS"*から最小汎化集合 *MGS* を抽出すると，{<$(1)[$(2)(3)(5)]$(5)>}が得られる．これを，文字列に戻すと，{<*A*[*B*|*BD*|*C*|*BB*]*C*>}（ただし，文字列間を区別するために文字列間に|を入れている）である．

4.5. 例

例えば，ミスマッチクラスタ *MIS*={<ロサンゼルス>,<ロサンジェルス>, <ロスアンゼルス>,<ロスアンジェルス>}から最小汎化集合を抽出することを考える．

このミスマッチクラスタをアラインメント処理により文字列長を整えると *MIS'*={<ロサーンゼールス>,<ロサンジェルス>, <ロスアンゼールス>,<ロスアンジェルス>}となる．

次に，*MIS'*を$変換により$(*n*)の記号列に変化すると，*MIS"*={< $(1) $(2) $(4) $(5) $(7) >,< $(1) $(2) $(4) $(6) $(7) >,< $(1) $(3) $(4) $(5) $(7) >,< $(1) $(3) $(4) $(6) $(7) >}が得られる．ここで，$(1) = <ロ>，$(2) = <サ>，$(3) = <スア>， $(4) = <ン>，$(5) = <ゼ>， $(6) = <ジェ>，$(7) = <ルス>である．

MIS" から最小汎化集合を抽出すると，*MGS*={<$(1)[$(2)$(3)]$(4)[(5)(6)]$(7)>}となる．これを元の文字列に直すと，{<ロ[サ|スア]ン[ゼ|ジェ]ルス>}が得られる．最小汎化集合から“ロ”の“ン”の間に“サ”と“スア”が入り，“ン”と“ルス”の間に“ゼ”と“ジェ”が入るという規則性を取り出すことができている．

5. 評価実験

提案手法の評価を，アミノ酸配列データベースに対して行った曖昧な問合せから得られたミスマッチクラスタから最小汎化集合を抽出することで行った．実験では文字列長が異なる部分文字列から構成されているミスマッチクラスタを正しく汎化できることと，抽出された最小汎化集合の汎化配列パターンに意味のあるパターンが含まれているかどうかを確認することによって，提案手法を評価する．

5.1. データセット

実験評価に用いられた Zinc Finger データセット(登録番号:PS00028) は PROSITE[5]から取得した．Zinc Finger データセットのデータ件数は 1839 件である．

実験では, 最初に, Zinc Finger データセットに対して，曖昧な問合せを行う．検索文字は<*C* - *x*(2,4) - *C* - x(3) - *L* - *x*(8) - *H* - *x*(3,5) - *H*>で，許容誤差数は 1 とした．*x*(2,4)は 2 文字～4 文字のワイルドカード領域（ワイルドカードとはどんな文字でもよいことを示す), x(3)は 3 文字のワイルドカード領域であることを示す.

曖昧な問合せにおける検索文字は各データセットに含まれるモチーフの一部分を用いている．この検索文字列に可変長ワイルドカード領域が含まれる場合は，それを固定長ワイルドカード領域が含まれる部分文字列の集合に表現し，その集合内の要素を論理和で結合した条件で問合せを行っている．

次に，曖昧な問合せによって得られたミスマッチクラスタから提案手法を用いて最小汎化集合を抽出する．ミスマッチクラスタを構成する部分文字列の件数，部分文字列の長さの特徴を表 1 に示す.

表 1 ミスマッチクラスタの特徴

データセット名	部分文字列の最小長	部分文字列の最大長	ミスマッチクラスタを構成する部分文字列の件数
Zinc Finger	21	25	14513

5.2. 実験結果

表 2 に実験結果を示す．14513 件の部分文字列から構成されるミスマッチクラスタから 111 件の要素から構成される最小汎化集合を抽出することができた．また，111 件の汎化配列パターンから逆にすべてのインスタンスを求めると，14513 件のインスタンスを求めることができることも確認した．このことから，文字列長が異なるインスタンスから構成されているミスマッチクラスタから最小汎化集合が正しく抽出できたといえる．

例えば，< *C* – [*x*(2) | *x*(4)] – *C* - *x*(3) - [*C*|*S*|*T*|*Y*|*H*|*G*|*V*|*F*|*L*] - *x*(8) - *H* - *x*(3,4) - *H* > は，抽出された最小汎化集合の汎化配列パターンの 1 つである．このパターンは，21 文字～24 文字の部分文字列から抽出された汎化配列パターンである．また，[*x*(2) | *x*(4)] と *x*(3,4)のように異なる長さのワイルドカード領域を示す文字列を 1 つの曖昧文字ドメインにまとめることができている. また, [*x*(2) | *x*(4)]は，*x*(2) もしくは *x*(4)であることを示す.

表 2 実験結果

データセット名	最小汎化 集合の要素数
Zinc Finger	111

5.3. 考察

抽出された最小汎化集合の各汎化配列パターンを支持数でランキングを行った．表 3 にランキング結果を示す．支持数とは，何本の配列に汎化配列パターンが含まれるかを示す値である．また，*EVAL*(汎化配列パターン)のうち，モチーフに該当するインスタンスがどれだけの割合で含まれるかも示した．すなわち，この割合は，｛モチーフに該当するインスタンス数｝÷｛*EVAL*(汎化配列パターン)のインスタンス数｝で計算した．

表 3 の割合を見てみると，上位 10 件のパターンすべてに関して，各汎化配列パターンに，50%以上の割合でモチーフに該当するインスタンスが含まれていることが分かる．また，5 位にランキングされている<*C* - x(2,4) - *C* - *x*(3) - *F* - *x*(8) - *H* - *x*(3,5) - *H*>は，そのインスタンスのすべてがモチーフに該当するインスタンスとなった．提案手法を用いることでモチーフ表現を抽出することができた．

さらに，Zinc Finger モチーフの配列パターンは，<*C* - x(2,4) - *C* - *x*(3) - [*LIVMFYWC*] - *x*(8) - *H* - *x*(3,5) - *H*>として知られている．上位 10 件のパターンすべてが，[*LIVMFYWC*]の曖昧文字の部分が異なるだけで，Zinc Finger モチーフの配列パターンに非常に似ていることが分かる．

次に，5 位にランキングされた汎化配列パターンをインスタンス分割して，インスタンス毎に支持数を求めた結果を表 4 に示す．

ただし，例えば，<*C*- *x*(2) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(3) -*HH* >という部分文字列の支持数をカウントする場合，<*C*- *x*(2) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(3) -*H*>と<*C*- *x*(2) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(4) -*H*>の両方の支持数がカウントされてしまうため，この結果はそれぞれ重複するインスタンスが出現する．したがって，支持数の合計は元の汎化配列パターンの支持数よりも大きくなる．

表 4 から分かるように，全文字列 9 件の内 4 件は 100 件以下の支持数であり，支持数でランキングすると下位になる．しかしながら，汎化処理により 1 つの汎化配列パターンとしてまとめることで，ランキングを上位にあげることができる．

表 3 最小汎化集合の汎化配列パターンのランキング（Zinc Finger データセット）

No	汎化配列パターン	支持数	割合
1	<*C*- [*x*(2)| *x*(4)] -*C*- *x*(3) -[*C*|*T*|*Y*|*F*|*L*]- *x*(8) -*H*- *x*(3,5) -*H*>	1751	80
2	<*C*- *x*(2) -*C*- *x*(3) -[*C*|*I*|*S*|*T*|*Y*|*H*|*A*|*G*|*M*|*V*|*F*|*L*]- *x*(8) -*H*- *x*(3,5) -*H*>	1731	58
3	<*C*- [*x*(2)| *x*(4)] -*C*- *x*(3) -[*C*|*S*|*T*|*Y*|*H*|*G*|*V*|*F*|*L*]- *x*(8) -*H*- *x*(3,4) -*H*>	1690	55
4	<*C*- *x*(2,4) -*C*- *x*(3) -[*S*|*T*|*Y*|*F*]- *x*(8) -*H*- *x*(3,4) -*H*>	1684	50
5	<*C*- *x*(2,4) -*C*- *x*(3) -*F*- *x*(8) -*H*- *x*(3,5) -*H*>	1678	100
6	<*C*- *x*(2) -*C*- *x*(3) -[*C*|*I*|*R*|*N*|*S*|*D*|*T*|*Y*|*H*|*A*|*W*|*G*|*M*|*V*|*F*|*L*]- *x*(8) -*H*- *x*(3,4) -*H*>	1667	50
7	<*C*- [*x*(2)| *x*(4)] -*C*- *x*(3) -[*C*|*I*|*S*|*T*|*Y*|*H*|*G*|*V*|*F*|*L*]- *x*(8) -*H*- *x*(3) -*H*>	1612	60
8	<*C*- *x*(2,3) -*C*- *x*(3) -[*S*|*T*|*Y*|*A*|*F*]- *x*(8) -*H*- *x*(3,4) -*H*>	1609	60
9	<*C*- *x*(2,3) -*C*- *x*(3) -[*S*|*F*]- *x*(8) -*H*- *x*(3,5) -*H*>	1593	50
10	<*C*- *x*(2,4) -*C*- *x*(3) -[*C*|*S*|*T*|*Y*|*F*]- *x*(8) -*H*- *x*(3) -*H*>	1578	60

表 4 <*C*- *x*(2, 4) −*C*- *x*(3) −*F*- *x*(8) −*H*- *x*(3, 5) −*H*> に対する各インスタンスの支持数

インスタンス	支持数
<*C*- *x*(2) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(3) -*H*>	1515
<*C*- *x*(2) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(4) -*H*>	541
<*C*- *x*(2) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(5) -*H*>	176
<*C*- *x*(3) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(3) -*H*>	39
<*C*- *x*(3) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(4) -*H*>	28
<*C*- *x*(3)-*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(5) -*H*>	1
<*C*- *x*(4) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(3) -*H*>	437
<*C*- *x*(4) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(4) -*H*>	162
<*C*- *x*(4) -*C*- *x*(3) -*F*- *x*(8)-*H*- *x*(5) -*H*>	23

次に，従来の手法よりもより意味のあるパターンを抽出できていることを示すために，従来の手法との比較を行う．従来の手法では，文字列長の異なるミスマッチクラスタから最小汎化集合を求めることができないため，9 個のミスマッチクラスタに分けて汎化を行い，結果を 1 つにまとめた．

従来の手法で抽出された最小汎化集合の各汎化配列パターンを支持数で同様にランキングを行い，インスタンスの割合も示した．表 5 にランキング結果を示す．

表 5 の割合を見ると，上位 10 件のパターンのほとんどが，50%を下回っていることが分かる．この結果と表 3 の提案手法の割合とを比較すると，本実験においてモチーフの抽出精度が向上できたということが言える．

また，表 5 から分かるように，たとえば 4 位にランキングされている，<*C*- *x*(2) - *C*- *x*(3) - *L* - *x*(8) - *H*- *x*(5) -[*ADEFGHIKLMNPQRSTY*]>など，Zinc Finger モチーフの配列パターンとは遠い表現が上位 10 位に入っていることが分かる．

さらに，従来の手法では上位 10 位からは，Zinc Finger モチーフは抽出されなかった．そして，1 位は 1500 件以上だがそれ以外はほとんど 500 件以下と，支持数が提案手法と比較して小さいことが分かる．

この結果から，提案手法を用いることでモチーフ表現を抽出することができ，汎化処理により 1 つの汎化配列パターンとしてまとめることで，ランキングを上位にあげることができたということが言える．

表 5　従来の手法で抽出された最小汎化集合の汎化配列パターンのランキング（Zinc Finger データセット）

No	汎化配列パターン	支持数	割合
1	<*C*- *x*(2) -*C*- *x*(3) -[*ACDFGHILMNRSTVWY*]- *x*(8) -*H*- *x*(3) -*H*>	1535	43
2	<*C*- *x*(2) -*C*- *x*(3) -[*ACDEFGHIKLMNPQRSTVWY*]- *x*(8) -*H*- *x*(4)- *H*>	573	40
3	<*C*- *x*(4) -*C*- *x*(3) -[*CFGHILSTVY*]- *x*(8) -*H*- *x*(3) -*H*>	442	60
4	<*C*- *x*(2) -*C*- *x*(3) - *L* - *x*(8) -*H*- *x*(5) -[*ADEFGHIKLMNPQRSTY*]>	203	35
5	<*C*- *x*(2) -*C*- *x*(3) -[*ACEFGHILMSTVY*]- *x*(8)- *H*- *x*(4) - *H*>	202	53
6	<*C*- *x*(4) -*C*- *x*(3) -[*ACFGHLQSTVY*] - *x*(8) -*H*- *x*(4) -*H*>	173	45
7	<[*CDEFGHKLNPQSTVWY*]- *x*(4) -*C*- *x*(3) - *L* - *x*(8) -*H*- *x*(3) -*H*>	167	37
8	<*C*- *x*(2) -*C*- *x*(3) -*F*- *x*(8) -[*CHLQTY*]- *x*(3) -*H*>	156	66
9	<*C*- *x*(2) -[*CGRTVWY*]- *x*(3) - *L* - *x*(8) -*H*- *x*(3) -*H*>	151	57
10	<[*CDEFHIKLNPRTVY*]- *x*(4) -*C*- *x*(3) - *L* - *x*(8) -*H*- *x*(4) -*H*>	71	42

6. まとめ

本論文では，ミスマッチクラスタに対してアラインメント処理を行い，アラインメント結果から最小汎化集合を抽出する手法を提案した．アラインメント処理を行うことで，異なる文字列長の部分文字列の最小汎化集合を抽出することが可能となった．

評価実験を行い文字列長が異なる部分文字列から構成されているミスマッチクラスタから最小汎化集合を抽出できることを確認した．また，最小汎化集合の各汎化配列パターンの支持数・割合を計算することにより，モチーフの抽出精度の向上が確認できた．

これからの課題として，類似性の低いテキストデータを検索対象とした場合，類似性が低い部分文字列が検索結果として得られるため，マルチプルアラインメントの精度を改善することや， $変換の方法の工夫などが今後の課題といえる．

謝辞

本研究の一部は，日本学術振興会，科学研究費補助金(基盤研究(C)，課題番号：20500137)，文部科学省・科学研究費補助金（課題番号：20700095）の支援により行われた．

参考文献

[1] 荒木 康太郎，田村 慶一，加藤 智之，北上 始：ミスマッチクラスタに対する最小汎化パターン抽出方式，日本データベース学会論文誌(DBSJ Letters)，Vol.6，No.3，pp.5-8，2007.

[2] K.Araki, K.Tamura, T.Kato, Y.Mori and H.Kitakami: Extraction of ambiguous sequential patterns with least minimum generalization from mismatch clusters, THE THIRD INTERNATIONAL CONFERENCE ON SIGNAL-IMAGE TECHNOLOGY and INTERNET-BASED SYSTEMS, IEEE Computer Society Press, pp. 32-39 (2007).

[3] H.Kimura, H.Kitakami, K.Araki and K.Tamura: A stepwise generalization method for extracting minimum generalized set from mismatch cluster, Proceedings of the 2008 International Conference on Bioinformatics and, Computational Biology (BIOCOMP'08), Vol.II, pp. 998-1004 (2008).

[4] 田村 慶一，木村 浩明，荒木 康太郎，北上 始: 段階的一般化法によるミスマッチクラスタを表現する最小汎化集合の効率的抽出，電子情報通信学会論文誌 D「データ工学特集号」, Vol.J93-D,No.3, pp.189-202,2010 年 3 月.

[5] http://kr.expasy.org/prosite/.

[6] 阿久津達也：バイオインフォマティクスの数理とアルゴリズム，共立出版，2007.