7E－3

図書目録データベースにおける近似索引技法を用いた冗長性除去の高速化

遠里由佳子

九州工業大学

1　はじめに

重複レコードの除去とは，データベースにおける同一の内容を持つ重複したレコードを一つにまとめ，冗長性を除去することである．これは，記憶容量の削減と検索効率の向上に有効である．そのため，重複レコードの除去に関して多くの研究が行なわれている [1]．しかし，データ作成時のデータ入力の誤りや記法の不統一による記載のゆらぎを持つデータベースに対しては，従来，研究されてきたような高速な除去アルゴリズムは直接には適用できない．そこで本稿では，レコードの近似的同一性を考えた場合に適用可能な高速な索引技法を提案する．この技法は，索引として故意に衝突を起こすハッシュ関数と，索引技法によって構成される．図書目録データを対象として実験を行ない，有効性を実証する．

2　図書目録データにおける複本関係

本稿では，冗長データの一例として，図書目録データベースにおける複本関係について考える．図書館の蔵書に複数個の同一出版物がある場合，最初の1冊を原本，残りの図書を複本と呼び，これらに対応する目録データをそれぞれ原本データと複本データと呼ぶ．九州工業大学の図書目録データベースでは，図書の新規登録の際にデータ入力者が気づいたごく少数の複本情報だけが手作業で入力され，ほとんどの複本データは手つかずで残されている．表1は，九州工業大学の図書目録データベースにおける複本データの例である．

表 1: 複本データの例

書名：データベース(NO 1) 著者：ホップクロフト,J. 図書番号: 1	書名：データ－ベース_NO 1_ 著者：ホップクロフト,_J 図書番号： 2

図書目録データベースにおいては，ほとんどの場合，書名・著者・出版社の欄の値が同じである図書レコードは複本関係にあると見なして良い．しかし，作成時に起きた入力ミスや記法のゆらぎによって，元来全く同じ書名と，著者，出版社を持つ複本データであっても，図書目録データベースのレコードとしては，全く同じとは限らない．本稿では，入力ミスなど簡単な記載のゆらぎのみを扱うこととし，次のように図書レコードの近似同一性を定義する．

3　近似同一性

データベースの属性を $A_1,\cdots,A_n$ とする．組 $[A_1 : V_1,\cdots,A_n : V_n]$ $(V_i \in \Sigma^*)$ をレコードといい，レコードの並び $DB = (R_1,\ldots,R_N)$ をデータベースという．

本稿では，文字列間の近似照合に基づいてレコードの近似同一性を定義する．正整数 d に対して，文字列 p,q が編集距離 d で同一であるとは，文字の挿入および，削除，置換の高々 d 回の適用で，p を q に変換できることをいい，$p \simeq_d q$ と書く．この関係 $\simeq_d$ は近似文字列照合 [2] によって効率よく判定可能である．

次に，データベース DB において，比較に用いる属性，例えば図書目録データベースにおける，書名，著者，出版者を $B_{k_1},\ldots,B_{k_m}$ とし，各属性に対応づけられた編集距離の組を $\delta = (d_{k_1},\ldots,d_{k_m})$ とする．このとき，2つの図書レコード

$$R = [A_1 : U_1,\cdots,A_n : U_n]$$
$$S = [A_1 : V_1,\cdots,A_n : V_n]$$

は，任意の $1 \le i \le m$ において，$U_{k_i} \simeq_{d_{k_i}} V_{k_i}$ が成立するならば，近似的に同一であるといい，$R \simeq_\delta S$ と書く．

4　重複除去問題

重複除去は，与えられたデータベースにおいて，2つのレコードを順に取り出して比較し，一方が他方に近似的に同一であるならば，片方を原本としてデータベースに残し，もう一方を取り除くことを繰り返しておこなう．しかし，上で導入した近似同一性は同値関係ではないので，比較の順序によって，除去されるレコードが異なる．そこで本稿では，重複除去をつぎの手続きで逐次的に計算されるものに限って考察する．この手続きは，入力としてデータベース $DB = (R_1,R_2,\cdots,R_N)$ を受けとり，出力として重複が除去されたデータベース $DB' = (R_{i_1},R_{i_2},\cdots,R_{i_{N'}})$ を計算する．

NORMALIZE
begin
　for $i := 1,\cdots,N$ **do**
　　for $j := i+1,\cdots,N$ **do**
　　　if $R_i \simeq_\delta R_j$ **then** DB から R_j を除去する;
　DB を出力する;
end

上のアルゴリズムは，重複データ数の多少にかかわらず，$(1/2)N^2$ 回のレコード比較を行なう．一般に，レコード数 N は非常に大きく，比較毎にレコードをディスクから読み込む必要があるので，この計算には膨大な計算時間を要する．

5　従来の索引技法の問題点

前節で述べた基本的な重複除去アルゴリズムは，一つのレコード R_i に対してそれと近似同一なレコードが存在するかどうかを検査するために，データベース全体を読み，レコードを比較しなければならない．このような場合に，ファイルの一部分を読み込むだけでこの検査をおこなうための方法に索引技法がある．比較に使う属性

Approximate indexing method for efficient elimination of duplications from bibliographic databases,
Yukako Tosato, Kyushu Institute of Techology

を B （キー属性）としたとき，索引技法では，各レコード R_i に対して，属性 B の値 v_i （キー）とレコード位置 i の組 (v_i, i) を関連付け，この組を集めてキーの値でソートして索引ファイルを作る．あるレコード R_i に対してそれと同一なレコードが存在するかどうかを検査するときは，2分探索などを用いて わずかな回数のキーの比較だけで，索引ファイルから $v_i = v_j$ となる組 (v_j, j) だけを見つけだし，レコード R_j を読み込んで比較できる．

しかし，レコードの比較に厳密な同一性でなく近似同一性を用いる場合には，索引技法は直接適用できない．なぜならば，文字列の同一性として近似同一性を採用しているため，$v_i \simeq_\delta v_j$ となる v_j を見つけるには，索引ファイル全体をはじめから調べる必要があるためである．例えば，文字列の順序として辞書式順序を用いているときには，最初の一文字が違えば索引ファイル中での位置はまったく変わってしまう．したがって，近似同一性に適用可能な新しい索引技法が必要となる．

6 近似索引技法

本節では，衝突をおこすようなハッシュ関数を用いて索引を構成し，索引ファイル全体を読み込まずに近似同一なレコードを検出する方法を提案する．

以下では，比較に使うキー属性が一つの場合について説明する．まず，キーを変換する関数について説明する．キー属性 B に関する編集距離を d とする．文字列から自然数への関数 f で，2つの文字列 u, w が近似的に同一ならば，対応する関数値の差 $f(u) - f(w)$ の絶対値がある定数 e 以内におさまるようなものを考える．この関数 f を近似索引関数と呼び，e を最大距離という．キー u に対する関数の値 $f(u)$ を近似キーと呼ぶ．索引ファイルとしては，組 (v_i, i) の代わりに，近似キーの値とレコード位置の組 $(f(v_i), i)$ を集めて，近似キー $f(v_i)$ の値でソートしたものを用いる．

あるレコード R_i に対してそれと同一なレコードが存在するかどうかを検査するときは，まず R_i を読み込み，そのキー v_i から近似キー $x = f(v_i)$ を計算する．このとき，R_i と近似同一になる可能性のあるレコードを R_j とし，そのキーを v_j とすると，$v_i \simeq_\delta v_j$ となるので，これに対応する組 (y, j) は，近似キー $y = f(v_j)$ が $x - e$ 以上 $x + e$ 以下であるような組だけであると分かる．索引ファイルは近似キーの値でソートされているから，このような組は索引ファイルの連続した一部分を占めることになる．キー属性が複数あるときは，第1キーで制限した後で，第2キー，第3キーで順に制限すれば良い．

以上の観察にもとづいて，改良した重複除去アルゴリズムをつぎに示す．入力データベースを $DB = (R_1, \cdots, R_N)$ とする．

NORMALIZE
begin
　索引ファイル $((x_1, i_1), \ldots, (x_N, i_N))$ を作成する;
　for $k := 1, \cdots, N$ **do**
　　$h := k + 1$;
　　while $x_h \leq x_k + e$ **do begin**
　　　if $R_{i_k} \simeq_\delta R_{i_h}$ **then**
　　　　DB から R_{i_h} を除去する;
　　　$h := h + 1$;
　　end
　DB を出力する;
end

近似索引関数 f としては，できるだけ最大距離 e が小さいものが良い．本稿で用いた近似索引関数 f と，編集距離 d に対応する最大距離 e はつぎのとおりである．

$f(u)$ ＝ u に出現する文字コードの総和．ただし，1byte 文字はそのまま加え，2byte 文字は下位 1byte の文字コードのみを加える

e ＝ $255 \cdot d$

7 実験

近似索引技法の有効性を確かめるために，九州工業大学附属図書館情報工学分館の図書目録データベースから複本をあまり含まない（全体の 3.7 %）データベース D1 と複本を多く含む（全体の 33.6 %）データベース D2 を抜きだし，実験をおこなった．近似同一性としては，書名が高々一文字違いであるものを考えた．実験は，レコード数を 50 冊から約 3000 冊まで 50 冊ずつ増加させ，冗長性除去にかかる CPU 時間をはかる方法で行なった．実験結果を図 1 に示す．

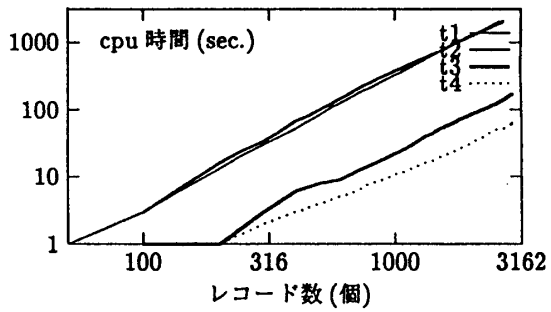

図 1: 実験結果．グラフ t1, t2 は，それぞれ，基本的アルゴリズムを D1, D2 に適用したもの．グラフ t3, t4 は，それぞれ，改良したアルゴリズムを D1, D2 に適用したもの．横軸と縦軸には，それぞれ，総レコード数と CPU 時間の 10 を底とする対数をとった．

図 1 からわかるように，計算時間のオーダーは共に $O(n^2)$ だが，係数に関しては改良したアルゴリズムの方が 15〜40 倍高速である．実際に，図書目録データ全体（レコード数 46300，ファイルのサイズ 8.7MB）に改良したシステムを適用したところ，従来のアルゴリズムでは 12 日間かかった重複除去が，30 分前後で実行できた．この重複除去によって，レコード数は元の 80.4 %に，ファイルサイズは 89.7 %に減少した．

8 おわりに

高速に冗長性除去を実行する近似索引技法を示し，その有効性を図書目録データベースを対象とした実験で検証した．この技法のより複雑な近似同一性への拡張や，近似同一性自体の例からの学習等が今後の研究課題である．

謝辞　日頃御指導いただいた篠原教授と有村助教授に感謝します．

参考文献

[1] Bitton D. and DeWitt, D. J., Duplicate record elimination in large data files. ACM Trans. Database Syst., vol. 8, no. 2, 255-265, 1983.

[2] Ukkonen, E., Finding approximate patterns in strings. J. Algorithms, vol. 6, no. 1, 132-137, 1985.