LD_002

重複レコード照合における分割統合照合方式の提案と有効性評価

Proposal and Evaluation of Divide and Merge Matching on Duplicated Records Detection

立石 健二† 久寿居 大†
Kenji Tateishi Dai Kusui

1．はじめに

重複レコード照合とは、DB 内の実質的に同じレコードをグルーピングすることをいう。表記は異なるが、人が見れば同じと判断できるレコード同士を実質的に同じとみなす。重複レコード照合は、例えば、異なる人/場所/方法によって管理された顧客 DB を統合する際のデータクリーニングに必要となる。重複レコード照合支援システムは、対象 DB が指定されると、重複レコードの候補をグループで表示する。利用者はそれぞれが真に重複であるかを確認し、重複レコードを判定する。

重複レコード照合支援システムが DB の重複候補を見つける時、データが大規模であれば調べる組み合わせも膨大になり、多大な時間がかかってしまう。従来の重複レコード照合は、精度は落ちるが処理の軽い手法（おおよその絞込み）と、処理は重いが精度が良い手法（詳細な絞込み）を組み合わせることによって行っていた[1,2,3]。例えば、レコードの重複を調べる場合に、Standard blocking[2]という方法ではデータの先頭から n 文字が一致するレコードをブロックにしたものを大量に作り、ブロックの中で詳細な絞込みを行う。また、Sorted Neighborhood Method(SNM)[3] では、特定のキーの値でレコードをソートした上で、各レコードの固定長ウィンドウの範囲内にある近接レコード群をブロックとして、その中で詳細な絞り込みを行う。詳細な絞込みは、例えば、編集距離や cosine 等で計算した類似度で絞込みを行う。

このような従来方式は、照合の精度と速度がトレードオフの関係にあり、大規模 DB になると照合速度を一定以下に保つために照合精度が低下してしまう問題があった。大規模 DB では、処理時間の増加を抑えるためにはおおよその絞込みにおいてより多くのレコードを絞り込む(ブロックを小さくする)必要があり、精度が低下してしまう。

そこで本稿では、おおよその絞りこみにおいて各ブロックを十分小さくなるまで階層的に分割し、詳細な絞りこみにおいて、階層的に近接するブロックに跨る重複レコード候補を統合する分割統合方式を提案し、上記のトレードオフの問題を解決する。

2．分割統合照合方式

本方式は、大きく 3 つの処理に分かれる。1) おおよその絞りこみにおいて各ブロックを一定な大きさ以下になるまで階層的に小さく分割するサブブロックへの分割処理(2.1 節)と、2) 詳細な絞込みにおいてサブブロック内の重複レコード候補グループを検出する類似度計算処理(2.2 節)と、3) 階層的に近接するブロックに跨る重複レコード候補グループを統合する統合処理(2.3 節)を順次行う。1 番目の処理で照合時間の増大を抑え、3 つ目の処理により照合精度の低下を抑えることで、精度を保ちつつ速度の向上を図る。

† NECインターネットシステム研究所

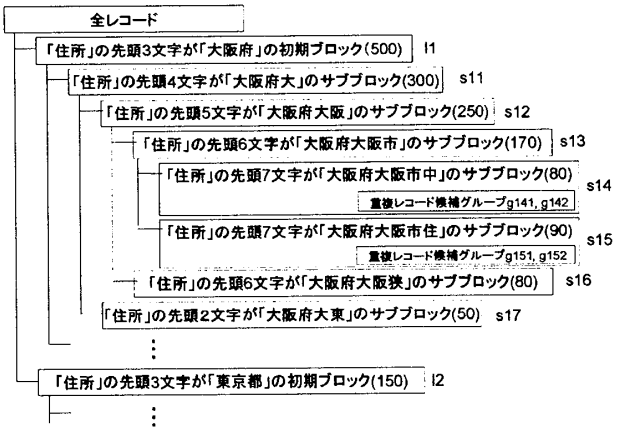

図 1 サブブロックへの分割処理

2.1 サブブロックへの分割処理

まず、Standard blocking[2]と同様に前方から n 文字をキーとしてレコードを初期ブロックに分割する。次に、キーのサイズを 1 文字ずつ増やしながら、階層的にサブブロックサイズの最小値以下になるまで分割する。図 1 では、初期ブロック I1 を、s11 から s17 のサブブロックに分割している。

2.2 サブブロック内での類似度計算処理

それぞれのサブブロック内のレコードペアを総当りで類似度計算し、閾値以上のレコードペアを一つのグループ(重複レコード候補グループ)にまとめていく。それぞれの重複レコード候補グループの任意の一つを代表レコードとする。類似度計算方法としては、例えば、編集距離や cosine を用いる。例えば、r1(レコード 1)と r2、および、r2 と r3 が重複レコード候補と判定された場合は r1 から r3 を 1 つの重複レコード候補グループとする。

2.3 重複レコード候補グループの統合処理

階層構造が深い(共通する文字列が多い)サブブロック群から順に親が共通するサブブロック群を一つにまとめる。この時、新たに同じサブブロックになった重複レコード候補グループをそれぞれの代表レコードを比較して統合する。例えば、図 1 の I1 の初期ブロックについては、最も階層の深い s14 と s15 を s13 へまとめる。この時、s14 と s15 の間の重複レコード候補グループの組み合わせ、すなわち、g141 と g151、g141 と g152、g142 と g151, g142 と g152 の統合可否をそれぞれの代表レコードを比較することで判断し、統合可能な場合はそれらを統合する。次に

s13 と s16 を s12 へまとめる...といった具合に I1 のサブブロックが一つにまとめられるまで繰り返す。

重複レコード候補グループの代表レコード同士を比較して類似度が閾値以上ならば統合することができる。しかし、代表レコード同士の類似度は閾値を越えてなくても、それらのグループに含まれる別のレコード同士が閾値を超えることは起こりうる。そこで、類似度閾値に2倍の余裕を持たせて準類似度閾値とし、代表レコード同士の類似度が準類似度閾値以下の場合は統合不可とする。例えば、類似度閾値が 90%の場合は準類似度閾値を 80%とする。類似度が準類似度閾値よりも大きく(>80%)、類似度閾値よりも小さい(<90%)場合は、代表レコード以外のレコード同士を比較して類似度閾値を超えるレコードペアが存在した場合は統合する。

3. 評価

3.1 実験方法

大規模 DB になると照合速度を一定以下に保つために照合精度が低下してしまう従来手法のトレードオフの問題を提案方式により解決できるかを評価した。

従来方式として用いたのは、Standard blocking [2]であり、データの先頭から n 文字をキーとしてブロックを生成し、ブロック内で重複レコードを検出する。重複判定のための類似度には、従来/提案方式共に編集距離を 0-1 に正規化した値を用いた。代表レコードは、各重複レコード候補グループの最も若い ID を持つレコードとした。システムの設定値として、重複候補の類似度閾値は 90%、サブブロックサイズの最小値を 100 レコードとした。使用したマシンの OS は Window2000、CPU は Pentium Xeon 2.6MHz、Memory は 3.4GByte を搭載している。

実験には、住所と名称の 2 フィールドを持つ 50 万レコードの顧客 DB を用いた。なるべく多くの重複候補を検出するため、まず、住所をキー文字列として従来/提案方式で重複レコード候補グループを求め、次に名称を初期ブロックとして同様に求め、最後に両者の和集合を求め、最終的な結果とした。その際、類似度計算は、住所・名称をあわせた全レコードを対象としている。

3.2 実験結果・考察

表 1 は提案手法、表 2 は従来手法の照合時間と削除レコード数(削除数)を示している。削除レコード数とは、重複レコード候補グループが全て重複であり代表レコード以外は全て削除すると判断した場合の削除レコードの総和を意味する。

提案手法と従来手法ともに、初期ブロックのキー文字列長を長くすると(初期ブロックを小さくして組合せ数を減らすと)照合時間は短くなるが、精度が下がる(削除レコード数が減る)傾向がある。

従来手法はキー文字列長が 5 文字の場合に照合時間は 6 時間以上かかっており、20 文字の場合には 5 文字の場合に比べて削除レコード数が 12%程度減少している。従来方式では、実際に何度か試して適切なキー文字列長(ブロックサイズ)を調整しなければならない。

一方で、提案手法はキー文字列を 2 文字にした(初期ブロックを大きくした)場合でも照合時間を約 13 分に抑えられている。またキー文字列長を長くしても従来手法ほど照合時間短縮の効果はない。したがって、提案手法ではキー文字列長の調製が必要なく常に短く設定すればよい。

表 1　提案手法

レコード数	初期ブロックキー文字列長	照合時間 提案手法	削除数 提案手法
500000	2 文字	13m28s	34380
500000	5 文字	4m45s	34366
500000	10 文字	2m26s	33500
500000	20 文字	46s	30767

表 2　従来手法

レコード数	初期ブロックキー文字列長	照合時間 従来手法	削除数 従来手法
500000	5 文字	6h44m51s	35860
500000	10 文字	38m15s	34912
500000	20 文字	2m27s	31701

提案方式でキー文字列長を 2 文字とし、従来方式では精度と速度のバランスが良さそうな 10 文字に調整したとしても、提案方式は従来方式よりも照合は約 3 倍速く、照合精度(削除レコード数)も約 1.5%の低下に抑えている。

以上の結果から、提案手法は大規模 DB でも初期ブロックの文字列長の調整なしに高速に照合でき、従来手法と同程度の重複レコードを見つけることができる(精度を保てる)ことがわかる。これは、大規模 DB になると照合速度を一定以下に保つために照合精度が低下してしまうトレードオフの問題が解決できたと考えてよい。

4. おわりに・今後の予定

本稿では、大規模な DB の重複レコード照合における精度と精度のトレードオフの問題の解消(精度を維持しつつ速度を向上すること)を目的として、おおよその絞りこみにおいて各ブロックを十分小さくなるまで階層的に分割し、詳細な絞りこみにおいて、階層的に近接するブロックに跨る重複レコード候補を統合する分割統合方式を提案した。50 万レコードの顧客 DB を用いた実験において、提案方式は照合時間を約 1/3 に抑え、従来方式と同程度の重複レコードを検出できることがわかった。

今回提案した分割統合照合は、先頭から n 文字をキーとして初期ブロックを作成しているので、先頭付近の文字列が異なる重複レコードは照合漏れになる可能性がある。しかしながら、今後 Suffix Array のようなデータ構造と組み合わせる[4]ことにより任意の位置からの n 文字をキーとした初期ブロックの作成およびサブブロックの分割統合が可能になり、上記の問題は解消できると期待する。

参考文献

[1] 相澤, 高須, 大山, 安達, "異種データベース間でのレコード照合に関する研究動向", NII Journal No.8, pp.43-51, 2004.

[2] M. A. Jaro, "Advances in Record Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida", Journal of the American Statistical Society, 84 (406), pp. 414-420, 1989.

[3] Mauricio A. Hernandez and Salvatore J. Stolfo, "The Merge/Purge Problem for Large Databases", SIGMOD 1995, pp.127-138, 1995.

[4] 相澤, 大山, 高須, 安達, "複数書誌データベース統合における重複エントリーの高速検出法", 情報処理学会研究報告 2004-FI-75, pp.111-118, 2004.