*Tạp chí Tin học và Điều khiển học, T. 13, S. 1 (1997) (31–41)*

# TÌM KIẾM TRONG HỆ CƠ SỞ DỮ LIỆU TÊN HỌ LATIN

PHƯƠNG MINH NAM

**Abstract.** The purpose of this paper to examine the first-name and family-names indentification problem in database for persons' management. Some algorithms are proposed, based on errors probability of occurrences of alphabets and frequence of occurrences of alphabets in the first and family-names in database. It is shown that these algorithms are very effective in application.

Khi xây dựng phần đảm bảo toán học cho nhiều hệ thống thông tin tự động hóa, thường gặp bài toán truy nguyên đồng nhất, nghĩa là xác định tính đồng nhất các mẫu tin của cùng một đối tượng.

Có hai kiểu đồng nhất cơ bản: Đồng nhất theo khóa và đồng nhất theo thông tin. Kiểu đồng nhất thứ hai nói chung là phức tạp và nó càng khó khăn khi những thông tin cơ sở để đồng nhất không chính xác và không chuẩn.

Thường bắt đầu xây dựng thuật toán truy nguyên đồng nhất cho vấn đề tìm kiếm: *với một mẫu tin đơn lẻ, phải tìm trong cơ sở dữ liệu xem có những mẫu tin nào của đối tượng trùng với đối tượng của mẫu tin đơn lẻ đó.*

Giả sử mẫu tin đơn lẻ là $X^*$, với các thuộc tính là $\{X_1, X_2, ..., X_n\}$.

Khi đó ta có thể viết $X^* = \{X_1, X_2, ..., X_n\}$.

Tương tự cho các mữa tin trong CSDL $Y^* = \{Y_1, Y_2, ..., Y_m\}$.

$X_1, X_2, ..., X_n$ (và tương ứng là $Y_1, Y_2, ..., Y_m$) ở đây là các thuộc tính mô tả của một người trong cơ sở dữ liệu.

Chẳng hạn:

$X_1$ = quốc tịch $= X.\text{qt}$

$X_2$ = tên họ $= X.\text{ht}$

$X_3$ = năm sinh $= X.\text{ns}$

$X_4$ = số hộ chiếu $= X.\text{hc}$

$X_5$ = số chứng minh thư $= X.\text{cm}$

...

Vấn đề đặt ra là lựa chọn cặp thuộc tính tương ứng $\{X_i\}$ và $\{Y_i\}$ nào để đặc trưng được cơ bản cho một người? lập hàm đo khoảng cách giữa chúng ra sao để xác định $X^*$ trùng với $Y^*$ hoặc $X^*$ thuộc vào lớp $Y^*$ với độ đo đã chọn.

Giải quyết bài toán cho trường hợp này hoặc cho trường hợp một số mẫu tin đơn lẻ đã phức tạp, song vấn đề đồng nhất trong các hệ thống liên kết tự động và số lượng bản ghi lớn thì là vấn đề càng đặc biệt phức tạp.

Hai vấn đề được lưu ý đặc biệt đây là việc xác định giá trị truy nguyên đồng nhất của từng thành phần thông tin và nghiên cứu phát hiện những quy luật sai sót để chế ngự nó.

Một trong những phương pháp nhằm cải tiến không ngừng thuật toán dựa trên mô hình automat tự học để liên tục hoàn thiện các thông số của metric, ngưỡng của mertic và ma trận sai sót.

Dưới đây trình bày thuật toán tìm kiếm (truy nguyên đồng nhất) đã được cài đặt cho hệ quản trị sơ sở dữ liệu về nhân sự với tên họ ghi bằng chữ Latin (viết tắt là CSDL-NSLT)

Từ kết quả thực tiễn đã chứng minh rằng những thuật toán được đề xuất là có hiệu quả và có thể giúp ích cho việc xây dựng các hệ thống thông tin tự động hóa tương tự.

## I - TÍNH ĐẶC THÙ VÀ PHỨC TẠP CỦA DỮ LIỆU

**1. Dữ liệu trong CSDL-NSLT được thu thập từ nhiều nguồn** khác nhau:

* từ các dạng có cấu trúc (dạng cột mục) như giấy tờ có giá trị căn cước di trú (chứng minh thư, hộ chiếu...);
* từ các dạng không cấu trúc như các thông tin thu được qua nghe phát âm, đọc qua điện thoại...

Dạng thông tin thứ hai rất phụ thuộc vào độ chuẩn xác, tính thận trọng và thậm chí cả vào trình độ của người cung cấp, người thu lượm thông tin.

Những thông tin quan trọng nhất liên quan tới việc xác minh (đồng nhất một người) bao gồm:

- Các chi tiết nhân thân: *tên họ, ngày tháng, năm sinh, quốc tịch, số hộ chiếu, thị lực, số chứng minh (hoặc số cá nhân / Identification)...*
- Ngoài ra còn có thể có thêm các chi tiết nhận dạng (ảnh), tự dạng (chữ viết)...

Thường 4 thuộc tính yếu tố *họ tên, ngày tháng năm sinh, quốc tịch và số hộ chiếu* ($X$.qt, $X$.ht, $X$.ns, $X$.hc) được dùng để truy nguyên đồng nhất và vấn đề được quan tâm là các thuật toán đồng nhất giữa các cặp tên họ.

### 2. Độ phức tạp của dữ liệu

Vì thu thập thông tin từ nhiều nguồn, lại không nhất quán như trên, làm cho dữ liệu trong CSDL-NSLT thêm đa dạng và phức tạp, cụ thể là:

• *Dữ liệu không đầy đủ, không nhất quán:* vì không phải quốc gia nào cũng có giấy tờ căn cước tùy thân và số chỉ tiêu thông tin trên đó giống nhau.

• *Ngôn ngữ không đồng nhất* người nước ngoài có thể sử dụng nhiều ngôn ngữ khác nhau trong các giấy tờ có giá trị căn cước di trú (hộ chiếu, thị lực, thẻ Id...) như: - ngôn ngữ theo hệ latin như tiếng Anh, Pháp, Tây Ban Nha, Đức...; ngôn ngữ đã qua phiên âm theo quy tắc phiên âm quốc tế (tiếng Trung Quốc, Nhật, Ả rập, Slavonic...); thói quen ghi tên họ, họ tên, viết tắt...

• *Số lượng có thể rất lớn và biến động:* trung bình hàng năm có tới hàng triệu bản ghi được tập hợp vào trong CSDL-NSLT.

• *Đặc điểm nhận dạng, tự dạng cũng rất khác nhau:* màu da, màu mắt, đặc điểm nhận dạng, tự nhận dạng khác.

• *Các sai sót ngẫu nhiên khác ảnh hưởng tới chi tiết nhân thân*

- Sai do nghe phát âm rồi ghi lại.
- Sai sót do bản thân người tự ghi vào các tờ khai, phiếu thu tin.

Tất cả những yếu tố trên đây tác động không nhỏ tới vấn đề tìm các thuộc tính và một thuật toán tốt, phù hợp thực tế cho bài toán đồng nhất, tìm kiếm trong CSDL-NSLT.

## II - THUẬT TOÁN TÌM KIẾM CÓ TRỌNG SỐ

Thuật toán truy nguyên đồng nhất theo trọng số được sử dụng trong các trường hợp:

• Xác minh một người, một nhóm người khi biết tên, họ...

• Đồng nhất các CSDL-NSLT để nhất thể hóa nhiều lần xuất hiện của một người.

Dưới đây, đưa ra hai thuật toán đã đề xuất sử dụng trong những năm qua ở hệ thống CSDL-NSLT, đồng thời so sánh sự độ chính xác và hiệu quả giữa chúng.

### 1. Thuật toán đồng nhất số 1 (No.1)

Thuật toán này được một số nước sử dụng để đồng nhất dữ liệu về người nhập xuất cảnh [3].

Giả sử mẫu tin đơn lẻ (để đồng nhất) là

$$X^* = \{X.\text{qt}, X.\text{ht}, X.\text{ns}, X.\text{hc}\}$$

Mẫu tin trong cơ sở dữ liệu là

$$Y^* = \{Y.\text{qt}, Y.\text{ht}, Y.\text{ns}, Y.\text{hc}\}$$

*1.1. Quy tắc tạo mã phát âm (codfonetics)*:

Để khắc phục tình trạng tên họ có sai sót, sai sót này do phát âm (rồi ghi lại) mà có từ thuộc tính $X$.ht tạo thêm thuộc tính mã âm cho họ (với người nước ngoài) và tên (cho người Việt Nam) kí hiệu là $X$.cod1.

Kí hiệu

$$\text{Mã\_01} = X.\text{qt} + X.\text{ns}$$

Quy tắc tạo mã phát âm (codefonectic) = của Họ (với người nước nước ngoài) hoặc tên (Việt Nam) như sau:

- Giữ nguyên chữ cái đầu tiên.
- Bỏ nguyên âm hoặc các phụ âm câm: A, E, O, U, Y, W, H.
- Gán mã số cho các nhóm chữ sau chữ cái đầu tiên

| | | |
|---|---|---|
| B, F, P, V | = | 1 |
| C, G, J, K, Q, S | = | 2 |
| M, N | = | 5 |
| D, T | = | 3 |
| X | = | 0 |
| R | = | 6 |
| L | = | 4 |

- Với các kí tự sau kí tự đầu
  - Thêm số 0 nếu không đủ phụ âm mã.
  - Nhiều phụ âm cùng nhóm chỉ giữ lại 1.
  - 2 chữ đầu đều giống nhau cũng chỉ giữ lại 1.
- Số hộ chiếu: Lấy số hộ chiếu của lần nhập cảnh cuối cùng.

Việc đồng nhất 2 mẫu tin được tiến hành theo quy tắc (QT):

QT1: hai mẫu tin được coi là trùng nhau (đồng nhất tự động) nếu
X.mã_01 = Y.mã_01 và X.cod1 = Y.cod1 và X.hc = Y.hc

QT2: hai mẫu tin được coi là gần nhau (xác minh) nếu
- X.mã_01 = Y.mã_01 và X.cod1 = Y.cod1 và
- "So mịn" phần tên (họ) tương ứng của X.cod1 và Y.cod1 đạt yêu cầu của "ngưỡng" (trường hợp này số hộ chiếu khác nhau).

Ngưỡng metric (sau đây viết tắt là Met) xác định như sạu:
- *3 kí tự đầu không được phép sai,*
- *các kí tự sau so theo từng nhóm 2 và được phép sai 1 chữ trong nhóm 2 chữ.*

*1.2. Thuật toán No.1:*

Sau đây là giải thuật mô phỏng

**1:** Đọc $X$, $Y$

**2:** So sánh $X$.mã_01 với $Y$.mã_01

a) nếu X.mã_01 = Y.mã_01 → chuyển tới **3:**

b) nếu không → chuyển tới **1:**

**3:** So sánh $X$.hc với $Y$.hc

a) nếu X.hc = Y.hc → chuyển tới **6:**

b) nếu không → chuyển tới **4:**

**4:** So sánh $X$.cod1 với $Y$.cod1

a) Nếu X.cod1 = Y.cod1 → chuyển tới **5:**

b) nếu không → chuyển tới **1:**

**5:** So sánh "mịn", so chi tiết tương ứng từng kí tự họ (của người nước ngoài) hoặc tên (của người Việt Nam) theo quy tắc:

*Gọi là "mờ" nếu:*

- Trùng nhau cả 3 kí tự đầu
- Các kí tự sau kí tự thứ 3 so sánh theo nhóm 2 và trong 2 kí tự thì được phép sai một

*a) nếu mờ, hiện lên màn hình hoặc in ra giấy để xác minh bằng mắt.*

*b) nếu không, chuyển lên bước 1:*

**6:** Đồng nhất tốt (hai mẫu tin/hai người là một).

*Đánh giá thuật toán No.1:*

• *Codfonetics chưa đủ mịn, chưa phản ánh đúng thực tế nên đã loại ra một số trường hợp thực chất là một nhưng khác codfonetics.*

• *Quy tắc không cho phép sai một trong 3 chữ đầu tiên loại đi khoảng 2,1% số đối tượng thực chất là một.*

• *Nhiều đối tượng khác nhau gộp làm một vì cùng mã_01, cùng tên.*

• *Việc tạo mã_01 buộc mỗi người phải có đủ quốc tịch, năm sinh, số hộ chiếu, điều này không phải lúc nào cũng đạt được.*

• *Với điều kiện của ta, không phải lúc nào cũng phân biệt rõ được đâu là họ, đâu là tên của người nước ngoài nên việc tạo codfonetics có khi không đạt đúng mục đích.*

• *Việc so sánh codfonetics được thực hiện theo luật "ngang nhau" nghĩa là so sánh toàn bộ số trùng codfnetics (không quan tâm đến tên họ có trùng nhau hay không) làm cho tốc độ tính toán rất chậm. Thực tế có tới 78% người xuất hiện*

*lần thứ 2 trong CSDL-NSLT trùng hoàn toàn tên tuổi với lần thứ nhất nhưng chưa được tận dụng.*

## 2. Ma trận sai sót và thuật toán đồng nhất số 2 (No.2)

Trên cơ sở thống kê thực tế, thấy sai sót trong việc nhập dữ liệu về chi tiết thân nhân có thể do các nguyên nhân sau:

- Ngôn ngữ khác nhau có cách viết khác nhau.
- Yếu tố tâm lý: mỏi mệt nên nhìn không rõ chữ, viết lẫn chữ, cuối từ viết ẩu hơn đầu từ.
- Yếu tố máy: sai sót ngẫu nhiên khi gõ trên bàn phím.

Như vậy sai sót không chỉ dừng lại ở mã phát âm nữa.

Bằng phương pháp thống kê toán học, sai sót giữa các chữ tuân theo quy luật số lớn và ổn định ở mức trên 17 ngàn phép thử (mẫu)

Chẳng hạn nhiều chữ có xác xuất lẫn rất lớn như:

$$U \to V,\ D \to O,\ A \to H,\ J \to I \ldots$$

*2.1. Ma trận sai sót:*

Thống kê trên tập gồm khoảng 25 ngàn mẫu [1], [2], được ma trận sai sót, trong đó các hàng và các cột là bảng chữ cái từ A đến Z (ứng với dãy số thứ tự 1 đến 26).

Các phần tử của ma trận kí hiệu là $P_{ij}$, biểu thị tần suất sai sót của chữ ứng với vị trí thứ $i$ sang vị trí thứ $j$ (trong bảng chữ cái).

Gọi tần số sai của chữ ở vị trí thứ $i$ sang vị trí thứ $j$ là $E_{ij}$.

Số cặp $(i, j)$ xuất hiện là $S_{ij}$.

Ta có công thức $\quad P_{ij} = \dfrac{E_{ij}}{S_{ij}}.$

Ví dụ: ANASTACIA xét 4 cặp (A,A)
ANHSTACIA (A,H)
(A,A)
(A,A)

$$P_{AA} = \frac{3}{4} \qquad P_{AH} = \frac{1}{4}$$

Ma trận sai sót phản ánh rất khách quan quy luật sai của các chữ và được lấy làm cơ sở để nhóm các chữ tạo thành codfonetics (được gọi là ma trận độ đo của sai lệch).

Ma trận sai sót được ứng dụng trong việc phân lớp các kí tự.

Sai sót còn phụ thuộc vào vị trí khác nhau của kí tự trong từ.

Thống kê trên 2 ngàn cặp tên họ có sai sót, lập được bảng tần xuất sai sót theo vị trí: từ vị trí 1-3, độ sai lệch giao động ở khoảng 5,1%, sai nhiều nhất ở các vị trí thứ 5, 6. Tần xuất sai sót theo vị trí được sử dụng trong việc chọn ngưỡng để "đo" sụ gần nhau của hai từ.

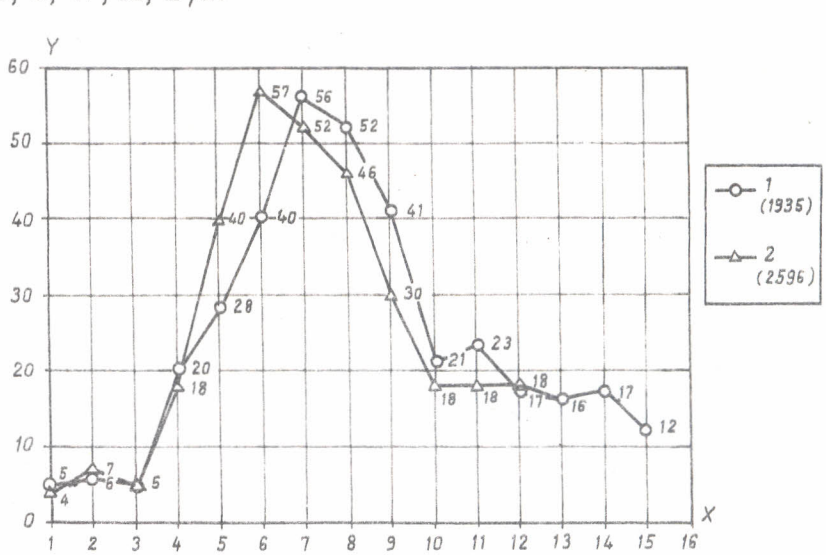
Gọi $V_i$ là tần xuất sai ở vị trí thứ $i$ của chữ trong các cặp từ có độ dài lớn hơn $i$, $E_i$ là sai sót ở vị trí thứ $i$, $S_i$ là tổng số từ có độ dài lớn hơn hay bằng $i$, ta có công thức

$$V_i = \frac{E_i}{S_i}$$

| | | | |
|---|---|---|---|
| Ví dụ: có 3 cặp | JTA | - ITA | |
| | ANAH | - ANDH | |
| | STOPHER | - STOPHFR | |
| thì | $V_1 = \frac{1}{3}$ | $V_3 = \frac{1}{3}$ | |
| | $V_2 = \frac{0}{3}$ | $V_4 = \frac{0}{2}$ | |
| | $V_5 = \frac{0}{1}$ | $V_6 = \frac{1}{1}$ | $V_7 = \frac{0}{1}$ |

Hình 1 biểu thị một đồ thị sai sót theo vị trí, rất có ý nghĩa trong việc chọn ngưỡng để xác định hai mẫu tin trùng nhau.

Sai sót còn phụ thuộc vị trí các kí tự trên bàn phím (chuẩn 101 phím). Sai số bàn phím thể hiện việc gõ nhầm giữa các phím lân cận nhau. Ví dụ nhóm ⟨M, N, K, J⟩; ⟨A, S, W, X, Z⟩...




*Hình 1.* Biểu thị sai sót theo vị trí

*2.2. Đồng nhất các từ trong thuộc tính tên họ:*

Tên họ được cấu thành bởi các từ , giả sử có hai từ là:

$$X.\text{ht} = \{A.\text{ht}_1,\ A.\text{ht}_2,\ A.\text{ht}_3,\ldots\ A.\text{ht}_n\}$$
$$Y.\text{ht} = \{B.\text{ht}_1,\ B.\text{ht}_2,\ B.\text{ht}_3,\ldots\ B.\text{ht}_n\}$$

Để đơn giản, đồng nhất các ký hiệu $A.\text{ht}_i$ với $A_i$ (tương ứng $B.\text{ht}_j$ với $B_j$). Xác định hàm đo độ gần giữa hai kí tự $A_i$, $B_j$ bất kỳ như sau:

$$\rho(A_i, B_j) = \begin{cases} 1 & \text{nếu } A_i \text{ thuộc lớp } B_j \\ 0 & \text{nếu } A_i \text{ không thuộc lớp } B_j \end{cases}$$

Ngưỡng để xác định khoảng cách giữa hai từ chọn như sau:

$$\delta = \begin{cases} \frac{1}{3} & \text{tổng số 3 kí tự đầu} \\ \frac{2}{5} & \text{tổng số kí tự sau kí tự thứ 3} \end{cases}$$

Khi đó hàm

$$\rho(X.\text{ht}, Y.\text{ht}) = \underbrace{\sum_{i=1}^{3} \rho(A_i, B_i)}_{\rho_1} + \underbrace{\sum_{i=4}^{n} \rho(A_i, B_i)}_{\rho_2}$$

là hàm đo khoảng cách giữa hai từ $X$, $Y$.

$X.\text{ht}$ và $Y.\text{ht}$ được coi trùng nhau nếu $\rho_1 + \rho_2 < \delta$.

Một cách trực giác: trong 3 kí tự đầu thì tối đa được phép sai 1 kí tự, các kí tự sau kí tự thứ 3 được phép sai tối đa $\frac{2}{5}$ tổng số kí tự (từ sau kí tự thứ 3).

Trong thực tế, việc cài đặt thuật toán có ứng dụng hàm đo độ sai sót gặp nhiều khó khăn:

- Tốc độ tính toán của các máy hiện có là PC nên rất chậm.
- Số lượng người nhập vào CSDL rất lớn, hàng triệu/năm.
- Để khắc phục một số nhược điểm của thuật toán No.1, cài đặt ứng dụng phù hợp môi trường thực tế, đã sử dụng ma trận sai sót và thuật toán phân loại tự động [4] trong việc phân lớp các kí tự, đưa ra quy tắc ghép nhóm và tạo codfonetics mới như sau:
  - Tất cả các từ trong tổ hợp tên họ đều được tạo codfonetics.
  - Kí tự đầu tiên cũng được nhóm và lấy đại diện. Nghĩa là, trong nhóm các chữ có độ nhầm lẫn sang nhau lớn, chọn một kí tự làm đại diện.
  - Các kí tự sau cũng được nhóm theo cách thức tương tự sau khi bỏ các nguyên âm và phụ âm câm.

*Quy tắc nhóm cụ thể như sau:*

**Ký tự đầu được phân nhóm và phân thành 8 nhóm:**

- Nhóm 1: K làm đại diện
- Nhóm 2 Z làm đại diện
- Nhóm 3: U làm đại diện
- Nhóm 4 I làm đại diện
- Nhóm 5: D làm đại diện
- Nhóm 6: A làm đại diện

- Nhóm 7: F làm đại diện - Nhóm 8: L làm đại diện

**Các kí tự sau chia làm 4 nhóm:**

- Nhóm 1: mã số 1 - Nhóm 2: mã số 2
- Nhóm 3: mã số 3 - Nhóm 4: mã số 4

Codfonetics lấy 4 kí tự (không phải 3 kí tự như No.1)

Dưới đây, mô tả tóm tắt thuật toán đồng nhất cho các đối tượng đơn lẻ, nghĩa là đồng nhất một mẫu tin với một tập mẫu tin "giống nó" về codfonetics.

*2.3. Thuật toán đồng nhất No.2:*

Từ thuộc tính $X$.ht tạo nên thuộc tính mã phát âm (tương ứng với các thành phần tên, đệm hoặc họ) kí hiệu là

$X$.cod 1, $X$.cod2, $X$.cod3

Kí hiệu

Mã_02 = $X$.qt + $X$.ht + $X$.ns + $X$.hc

Họ tên sắp xếp theo thứ tự cod1, cod2, cod3.

Sau đây là giải thuật mô phỏng:

**1:** Đọc $X^*$, $Y^*$

**2:** So sánh $X$.mã_02 với $Y$.mã_02

2.1: nếu $X$.mã_02 = $Y$.mã_02 → chuyển tới **6:**

2.2: nếu không → chuyển tới **3:**

**3:** So sánh $X$.cod$i$ + $X$cod$j$ với $Y$.cod$i$ + $Y$.cod$j$

3.1: nếu trùng → chuyển tới **4:**

3.2: nếu không → chuyển tới **1:**

**4:** So sánh "mịn" chi tiết từng kí tự của họ (hoặc tên đối với người Việt nam) theo quy tắc:

*Gọi là "mờ" nếu*

- 3 kí tự đầu ít nhất $\frac{2}{3}$ số kí tự trùng nhau (sai không quá 1 kí tự)
- các kí tự sau, có ít nhất $\frac{3}{5}$ số kí tự, trùng nhau (sai không quá 2 trong 5 kí tự)

4.1: nếu mờ → chuyển tới **5:**

4.2: nếu không → chuyển tới **1:**

**5:** So sánh $X$.ns với $Y$.ns

5.1: nếu trùng → chuyển tới **6:**

5.2: nếu không → chuyển tới **7:**

**6:** So sánh *X*.hc với *Y*.hc

6.1: nếu trùng → chuyển tới **8:**

6.2: nếu không → chuyển tới **9:**

**7:** So sánh số *X*.hc với *Y*.hc

7.1: nếu trùng → chuyển tới **9:**

7.2: nếu không → chuyển tới **1:**

**8:** Đồng nhất tốt

**9:** Xác minh.

*Đánh giá thuật toán No.2:*

Qua thực tế nhiều năm sử dụng, đạt một số hiệu quả sau:

• *Khắc phục về cơ bản các hạn chế của thuật toán No.1, nhất là đã tạo codfonetics đủ mịn, phản ánh đúng thực tế quy luật sai sót ngẫu nhiên; mặt khác tận dụng các trường hợp không đủ thông tin nhưng vẫn tìm kiếm để giúp việc xác minh được thực hiện trên các tập hạn chế về số lượng*

• *Tốc độ tìm kiếm rất nhanh đối với trường hợp trùng hoàn toàn tên họ, năm sinh, hộ chiếu;*

• *Đã đồng nhất hàng triệu trường hợp, độ chính xác cao (sai số dưới 1/300.000) Trên cơ sở đồng nhất hàng chục triệu người, đã phát hiện hàng chục ngàn trường hợp tên họ, hộ chiếu, năm sinh có sai sót nhưng vẫn đồng nhất được và với số mẫu đồng nhất này, thuật toán liên tục được bổ sung, hoàn chỉnh (tự học).*

• *Thuật toán trên đã được triển khai cài đặt trong hệ thống MTĐT Cục xuất nhập cảnh (Các trung tâm MTĐT ở Hà Nội, Đà Nẵng, Tp. Hồ Chí Minh, các trạm công an cửa khẩu sân bay quốc tế...); đơn vị quản lý xuất nhập cảnh công an các địa phương; một số đơn vị khác thuộc Bộ Nội vụ.*

## BẢNG TẦN SUẤT XUẤT HIỆN CỦA CHỮ CÁI TRONG VÙNG HỌ VÀ TÊN

| | **Người nước ngoài** 340.000 mẫu tin | 1.380.000 mẫu tin (17.535.194 kí tự) | **Người Việt Nam** 104.354 mẫu tin (1.252.984 kí tự) |
|---|---|---|---|
| A | 0,1436 | 0,1049 | 0,09806 |
| B | 0,0193 | 0,0112 | 0,00756 |
| C | 0,0210 | 0,0357 | 0,02191 |
| D | 0,0250 | 0,0192 | 0,02366 |
| E | 0,0748 | 0,0812 | 0,05190 |
| F | 0,0076 | 0,0064 | 0,00001 |

| | Người nước ngoài 340.000 mẫu tin | 1.380.000 mẫu tin (17.535.194 kí tự) | Người Việt Nam 104.354 mẫu tin (1.252.984 kí tự) |
|---|---|---|---|
| G | 0,0189 | 0,0471 | 0,07757 |
| H | 0,0559 | 0,0786 | 0,11482 |
| I | 0,0951 | 0,0842 | 0,05918 |
| J | 0,0132 | 0,0098 | 0,00003 |
| K | 0,0400 | 0,0250 | 0,00590 |
| L | 0,0444 | 0,0381 | 0,01590 |
| M | 0,0489 | 0,0298 | 0,02005 |
| N | 0,0721 | 0,1136 | 0,18562 |
| O | 0,0775 | 0,0594 | 0,04875 |
| P | 0,0148 | 0,0126 | 0,01341 |
| Q | 0,0005 | 0,0020 | 0,00709 |
| R | 0,0674 | 0,0443 | 0,01490 |
| S | 0,0568 | 0,0409 | 0,00350 |
| T | 0,0494 | 0,0450 | 0,06733 |
| U | 0,0410 | 0,0596 | 0,09415 |
| V | 0,0097 | 0,0089 | 0,02469 |
| W | 0,0119 | 0,0112 | 0,00003 |
| X | 0,0011 | 0.0012 | 0,00393 |
| Y | 0,0150 | 0,0253 | 0,04005 |
| Z | 0,0060 | 0,0037 | 0,00002 |

## TÀI LIỆU THAM KHẢO


1. Phương Minh Nam, *Phân tích hệ thống và phần mềm một hệ thông tin*, Báo cáo tổng kết đề tài NCKH cấp nhà nước, mã số 60-90, 1990.
2. Phương Minh Nam, *Đề án tổng thể "Xây dựng hệ thống thông tin - máy tính xuất nhập cảnh"*, 6/1993. Tài liệu lưu hành nội bộ.
3. Nguyễn Doãn Tiến, Trương Đức Thái, Phương Minh Nam, *Thuật toán truy nguyên đồng nhất*, Kỉ yếu tuần lễ tin học lần thứ 2, 1990.
4. Nguyễn Bá, *Thuật toán phân loại tự động Doropheuk*, Bài giảng lớp nghiên cứu sau đại học, ĐHBK Hà Nội.


*Trung tâm Thông tin Cục Xuất nhập cảnh*

*Bộ Nội vụ.*

*Nhận bài ngày 2-8-1996*