(12) **BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ**

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **1-0020552**

(51)[7] **G06F 17/27**, 17/273

(13) **B**

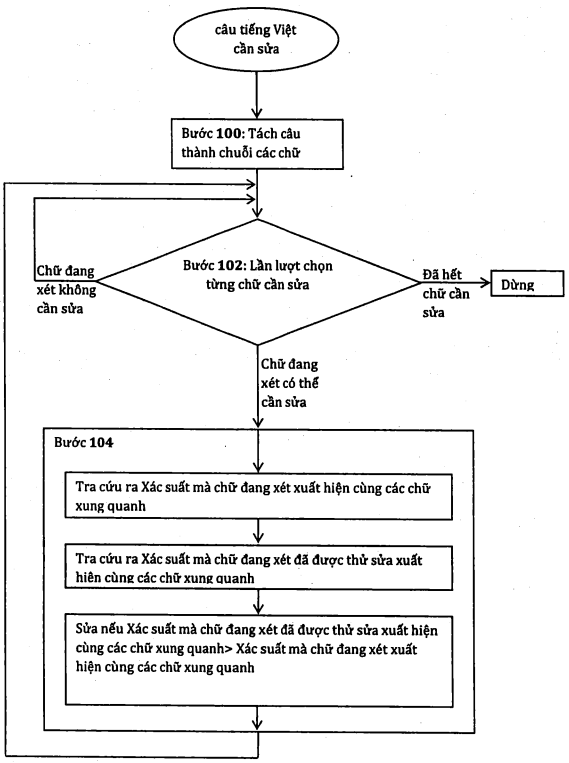
(21) 1-2014-01503
(22) 08.05.2014
(45) 25.02.2019 371
(43) 27.10.2014 319

(73) VIỆN NGHIÊN CỨU CÔNG NGHỆ FPT (VN)
Số 8 Tôn Thất Thuyết, huyện Từ Liêm, thành phố Hà Nội

(72) Nguyễn Tiến Dũng (VN), Phạm Thái Hoàng (VN), Trần Thế Trung (VN)

(54) VẬT GHI ĐỌC ĐƯỢC BẰNG MÁY TÍNH CÓ CHỨA CÁC LỆNH ĐƯỢC MÃ HÓA ĐỂ THỰC HIỆN QUY TRÌNH TỰ ĐỘNG SỬA LỖI LẪN LỘN GIỮA CHỮ L VÀ CHỮ N TRONG VĂN BẢN TIẾNG VIỆT

(57) Sáng chế đề xuất vật ghi đọc được bằng máy tính có chứa các lệnh được mã hóa để thực hiện quy trình tự động sửa lỗi văn bản tiếng Việt chứa những chữ bị ghi nhầm chữ cái đầu "1" thành "n" và ngược lại. Quy trình gồm bước xác định các chữ bắt đầu bởi chữ cái "1" hoặc "n" và các chữ xung quanh nó, tiếp nối bằng bước xác định xem liệu xác suất mà chữ này được sửa lại, thay chữ cái đầu từ "1" thành "n" hoặc ngược lại, đứng cạnh các chữ xung quanh, có cao hơn xác suất mà chữ gốc đứng cạnh các chữ xung quanh hay không, nếu có thì thực hiện việc sửa. Ngoài việc tận dụng thông tin của các chữ xung quanh chữ cần xem xét sửa, quy trình sửa lỗi có thể được mở rộng để tận dụng đặc trưng của tiếng Việt để xử lý nhanh những chữ bắt đầu bằng "n" và có chữ cái liền sau là "g", "h", hoặc tự động thu thập và sử dụng thông tin thói quen dùng từ của người dùng để làm tăng độ chính xác của việc sửa lỗi, mà không đòi hỏi người dùng phải chủ động cung cấp thêm thông tin gì.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến quy trình tự động sửa lỗi văn bản tiếng Việt chứa những chữ bị ghi nhầm chữ cái đầu "l" thành "n" và ngược lại.

Tình trạng kỹ thuật của sáng chế

Với xu hướng dịch chuyển tương tác giữa người dùng với máy tính sang dạng sử dụng ngôn ngữ nói tự nhiên, máy tính cần có khả năng hiểu được ngôn từ được đưa ra bởi người dùng, bao gồm cả những cách sử dụng ngôn từ không theo quy ước chuẩn.

Trong tiếng Việt, một dạng sử dụng ngôn từ không theo chuẩn, nhưng lại có độ phổ biến nhất định, là sử dụng nhầm "n" thành "l" hay ngược lại. Khi những người dùng có thói quen sử dụng nhầm như vậy nói chuyện với máy tính, giọng nói của người này vẫn có thể được chuyển hóa nguyên văn thành văn bản, thông qua các quy trình xử lý thông tin được biết đến rộng rãi trong lĩnh vực gọi chung là chuyển tiếng nói thành văn bản (*speech to text*). Những văn bản này trước khi được sử dụng cần được chuẩn hóa, tức là tự động sửa các chỗ nhầm thành cách dùng chuẩn.

Trên thế giới đã có rất nhiều công bố về các quy trình tự động sửa chữa các chữ sai chính tả. Đa phần các quy trình đã được công bố đều thực hiện ít nhất một công đoạn là tra các từ trong văn bản trong một từ điển. Ví dụ hồ sơ đăng ký sáng chế US 8176419 B2 và US 7076731 B2 của Microsoft cho thấy cách sử dụng vòng lặp để tra từ điển và xây dựng lại văn bản đúng chính tả giống nhất với một văn bản có lỗi chính tả đầu vào. Một số sáng chế khác tận dụng thêm thông tin ngữ cảnh, tức là những từ ngữ xuất hiện xung quanh từ đang được kiểm tra, để hỗ trợ cho việc tự sửa chính tả từ đang được kiểm tra. Ví dụ hồ sơ sáng chế US 8365070 B2 của Samsung ngoài việc xây dựng và sử dụng từ điển còn xây dựng và sử dụng cả cơ sở dữ liệu ngữ cảnh, trong việc tự động sửa chính tả.

Các sáng chế đã có trong lĩnh vực này áp dụng cho các trường hợp sai chính tả nói chung, nhưng chưa thực sự đạt hiệu quả cao nhất cho bài toán cụ thể là sai chữ cái "n" thành "l" và ngược lại. Lý do là:

thứ nhất, với bài toán cụ thể là sai chữ cái "n" thành "l" và ngược lại, số lượng các lựa chọn khi tìm cách tự động sửa là rất giới hạn; ví dụ với chữ "lầm", chỉ cần xem xét hai khả năng "lầm" và "nầm"; thay vì phải xem xét một dải rộng hơn các lựa chọn; nếu quy trình sửa tự động chỉ tập trung và số lượng nhỏ các lựa chọn, việc sửa tự động sẽ nhanh hơn, và nhiều khi chính xác hơn;

thứ hai, các từ tiếng Việt có tỷ lệ từ ghép bởi nhiều chữ khá cao, ví dụ "sai lầm" ghép bởi hai chữ, do đó việc tra từ điển chỉ một chữ, trong ví dụ này là chữ "lầm", mà không xét đến các chữ xung quanh sẽ cho hiệu quả thấp;

thứ ba, và là lý do quan trọng nhất, là việc dùng nhầm chữ cái "l" và "n" phụ thuộc rất lớn vào thói quen ngôn ngữ của cá nhân, do đó một quy trình có thể tự động sử dụng được thông tin lịch sử sử dụng ngôn từ của người dùng sẽ có khả năng sửa chính tả nhanh và chính xác hơn;

cuối cùng, một số đặc điểm rất đặc trưng của tiếng Việt có thể được tận dụng để làm nhanh quá trình xử lý thông tin, chẳng hạn với những từ bắt đầu bằng "n" và có chữ cái liền sau là "g", "h" thì có thể được coi là không cần sửa.

Trong các sáng chế đã được đề cập trên thế giới, việc tích hợp thông tin thói quen sử dụng ngôn từ của người dùng, một cách hoàn toàn tự động chứ không cần đến sự chú tâm của người dùng, chưa được quan tâm. Đơn đăng ký sáng chế US 4797855 A đã có ý tưởng sử dụng thông tin lịch sử của người dùng trong việc gợi ý những từ viết đúng chính tả. Tuy nhiên sáng chế này đòi hỏi người dùng phải chủ động nhập vào những lựa chọn đúng chính tả để máy móc ghi nhận lại những lựa chọn này và sử dụng các thông tin đã ghi nhận để hỗ trợ cho lần gợi ý từ đúng chính tả sau đó. Với bài toán cụ thể là sửa sai chữ cái "n" thành "l" và ngược lại, có thể hình dung đến một phương án hoàn toàn tự động thu được thông tin thói quen sử dụng ngôn từ của người dùng mà không đòi hỏi người dùng phải để ý và nhập thông tin vào máy móc.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là đề xuất vật ghi đọc được bằng máy tính có chứa các lệnh được mã hóa để thực hiện quy trình tự động sửa lỗi văn bản tiếng Việt chứa những cụm từ bị ghi nhầm chữ cái "l" thành chữ cái "n" và ngược lại, có khả năng tự động ghi nhận và sử dụng được thông tin về thói quen sử dụng ngôn từ của người dùng, liên quan đến "l" và "n", mà không cần gây sự chú ý của người dùng, không đòi hỏi người dùng phải chủ động cung cấp thêm thông tin gì.

Cụ thể quy trình tự động sửa lỗi văn bản tiếng Việt chứa những chữ ghi nhầm chữ cái đầu "l" thành "n" và ngược lại, từ một câu tiếng Việt đầu vào, gồm các bước:

phân tách câu đầu vào thành chuỗi các chữ; chẳng hạn nhờ các dấu cách;

lần lượt kiểm tra các chữ, theo một thứ tự nhất định, ví dụ từ đầu tới cuối chuỗi các chữ, xem có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và sau chữ cái "n" không phải chữ cái "g" hoặc "h" không, hoặc kiểm tra xem các chữ có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và chữ cái thứ hai sau "l" hoặc "n" là nguyên âm không;

với mỗi chữ thỏa mãn điều kiện kiểm tra ở bước trên thì thực hiện:

lấy ra chữ này, gọi là w_0, và a chữ đứng trước chữ w_0 trong chuỗi các chữ, gọi là w_{-a}, w_{-a+1}, ..., w_{-1}, và b chữ đứng sau w_0 trong chuỗi các chữ, gọi là w_1, w_2, ..., w_b, với a và b là hai số nguyên đã định trước, ví dụ $a = 1$ và $b = 1$; trong trường hợp không có đủ a chữ đứng trước w_0 hoặc không có đủ b chữ đứng sau w_0, thì các chữ còn thiếu được mặc định bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

đưa w_0 và w_{-a}, w_{-a+1}, ..., w_{-1} và w_1, w_2, ..., w_b vào một bảng tra đã lập sẵn để tra ra một con số tương ứng với bộ các chữ cho vào, gọi là $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0, w_1, w_2, ..., w_b)$; bảng tra này gọi là bảng B; chẳng hạn được xây dựng bằng cách thống kê lại từ một lượng lớn các câu tiếng Việt, trong đó:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với số lần xuất hiện chuỗi các chữ $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b$ chia cho số lần xuất hiện chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}$ trước một chữ nào đó rồi tiếp đến là chuỗi $w_1, w_2, ..., w_b$;

hoặc:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với:

số lần xuất hiện chuỗi các chữ $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0$ chia cho

số lần xuất hiện chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}$ trước một chữ nào đó nhân với

số lần xuất hiện chuỗi các chữ $w_0, w_1, w_2, ..., w_b$ chia cho

số lần xuất hiện chuỗi $w_1, w_2, ..., w_b$ sau một chữ nào đó;

trong phép thống kê để xây dựng bảng B như vừa nêu, khi không có đủ a chữ đứng trước hay b chữ đứng sau một chữ nhất định trong câu ở dữ liệu thống kê, các chữ bị thiếu cũng đặt bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

tạo ra tập hợp T, chứa tất cả các chuỗi được sinh từ chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b$ bằng cách thay thế một hoặc một số các chữ trong chuỗi này, thỏa mãn điều kiện kiểm tra ở bước thứ 2 của quy trình, bằng chữ có chữ cái đầu khác đi nhưng vẫn là "l" hoặc "n";

tra từ bảng B ra các giá trị $P(t)$;

nếu có chuỗi t thuộc T, thỏa mãn $P(t)\, f(\alpha_{"l"}, \alpha_{"n"}, \alpha_{t_1}, \alpha_{t_2}, ..., \alpha_{t_{a+b+1}}) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)\, f(\alpha_{"l"}, \alpha_{"n"}, \alpha_{w_{-a}}, ..., \alpha_{w_0}, ..., \alpha_{w_b})$, thì:

sửa toàn bộ chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b$ thành t;

tăng giá trị $\alpha_{"l"}$ thêm một giá trị bằng với số các chữ trong t khác với chuỗi gốc $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b$ vì bị thay thế chữ cái đầu từ "l" thành "n";

tăng giá trị $\alpha_{"n"}$ thêm một giá trị bằng với số các chữ trong t khác với chuỗi gốc w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b vì bị thay thế chữ cái đầu từ "n" thành "l";

tăng giá trị α_{t_i} thêm 1, cho mọi t_i, i thuộc 1, 2, ... $a+b+1$, khác với chữ tương ứng trong chuỗi gốc w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b;

với t_1 là chữ thứ nhất trong chuỗi t, t_2 là chữ thứ hai trong chuỗi t, ... t_{a+b+1} là chữ thứ $a+b+1$ trong t; $\alpha_{"l"}$, $\alpha_{"n"}$ và α_w, với w được dùng để chỉ một chữ nào đó, là tham số được lưu trữ khi quy trình được áp dụng lặp đi lặp lại riêng cho một người dùng nhất định, trong đó $\alpha_{"l"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "n" thành bắt đầu bằng "l"; $\alpha_{"n"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "l" thành bắt đầu bằng "n"; α_w có giá trị bằng số lần mà quy trình đã tự động sửa thành w; còn f là hàm số có $a+b+3$ biến, có giá trị bằng 1 khi các $\alpha_{"l"}$, $\alpha_{"n"}$ và α_w bằng 0, và tăng dần khi $\alpha_{"l"}$, $\alpha_{"n"}$ và α_w tăng.

Mô tả vắn tắt các hình vẽ

Hình 1 thể hiện sơ đồ khối của một phương án thực thi cơ bản của quy trình đề xuất bởi sáng chế;

Hình 2 thể hiện sơ đồ khối của một phương án thực thi của quy trình đề xuất bởi sáng chế, có ghi nhận và sử dụng thông tin lịch sử thói quen dùng từ của người dùng.

Mô tả chi tiết sáng chế

Phương án cơ bản

Hình 1 mô tả sơ đồ khối của một phương án cơ bản nhất cho quy trình được đề xuất bởi sáng chế này. Đầu vào của quy trình là một đoạn văn bản cần tự động sửa những chữ bị ghi nhầm chữ cái đầu "l" thành "n" và ngược lại.

Ở bước 100, văn bản đầu vào được phân tách thành các chữ, nhờ dấu cách, hoặc dấu ngắt câu (như dấu phẩy, dấu chấm phẩy).

Ở bước 102, lần lượt từ đầu tới cuối, kiểm tra xem các chữ có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và sau chữ cái "n" không phải chữ cái "g" hoặc "h" không. Cũng có thể, theo một phương án khác, là kiểm tra xem các chữ có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và chữ cái thứ hai sau "l" hoặc "n" là nguyên âm không.

Ở bước 104, với mỗi chữ thỏa mãn điều kiện kiểm tra ở bước 2, tức là, ví dụ, bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và sau chữ cái "n" không phải chữ cái "g" hoặc "h", thì thực hiện:

lấy ra chữ này, gọi là w_0, và a chữ đứng trước chữ w_0 trong câu đầu vào, gọi là w_{-a}, w_{-a+1}, ..., w_{-1}, và b chữ đứng sau w_0 trong câu đầu vào, gọi là w_1, w_2, ..., w_b, với a và b là hai số nguyên đã định trước, ví dụ $a = 1$ và $b = 1$; trong trường hợp không có đủ a chữ đứng trước w_0 hoặc không có đủ b chữ đứng sau w_0, thì các chữ còn thiếu được mặc định bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

đưa w_0 và w_{-a}, w_{-a+1}, ..., w_{-1}vàw_1, w_2, ..., w_b vào một bảng tra đã lập sẵn để tra ra một con số tương ứng với bộ các chữ cho vào, gọi là $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$; bảng tra này gọi là bảng B, được xây dựng bằng cách thống kê lại từ một lượng lớn các câu tiếng Việt nào đó, ví dụ từ hàng trăm triệu câu tiếng Việt có trong các bài báo điện tử, trong đó số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với số lần xuất hiện chuỗi các chữ w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b chia cho số lần xuất hiện chuỗi w_{-a}, w_{-a+1}, ..., w_{-1} trước một chữ nào đó rồi tiếp đến là chuỗi w_1, w_2, ..., w_b;trong phép thống kê để xây dựng bảng B như vừa nêu, khi không có đủ a chữ đứng trước hay b chữ đứng sau một chữ nhất định trong câu ở dữ liệu thống kê, các chữ bị thiếu cũng đặt bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

tạo ra chữ mới w_0', là chữ w_0 có chữ cái đầu khác đi nhưng vẫn là "l" hoặc "n";

tra từ bảng $BraP(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0' , w_1, w_2, ..., w_b)$;

nếu $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0', w_1, w_2, ..., w_b) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0, w_1, w_2, ..., w_b)$ thì sửa w_0 thành w_0'.

Phương án cơ bản nêu trên đã tập trung khai thác chỉ hai lựa chọn chính tả cho bài toán, tận dụng thông tin của các chữ xung quanh chữ cần xem xét sửa, và tận dụng đặc trưng của tiếng Việt để xử lý nhanh những chữ bắt đầu bằng "n" và có chữ cái liền sau là "g", "h". Tuy nhiên, phương án này chưa tận dụng thông tin thói quen sử dụng ngôn từ của người dùng.

Phương án sử dụng thông tin về thói quen của người dùng

Hình 2 mô tả sơ đồ khối của phương án có tự động thu thập và sử dụng thông tin thói quen dùng từ của người dùng, mà không đòi hỏi người dùng phải chủ động cung cấp thêm thông tin gì. Ý tưởng cơ bản là:

khi quy trình được lặp lại nhiều lần với những câu đầu vào từ cùng một người dùng, số lần cần sửa, hay không cần sửa sẽ phản ánh thói quen dùng từ của người dùng, rằng đó là người thường dùng đúng chính tả, hay đó là người thường nói nhầm "n" thành "l", hay đó là người thường nói nhầm "l" thành "n", ... tổng cộng có 4 loại người dùng;

xác suất phải sửa lại, hay không phải sửa lại, trong lần chạy tiếp theo của quy trình sẽ bị ảnh hưởng bởi phân loại người dùng; chẳng hạn xác suất phải sửa "l" thành "n" tăng lên nếu người dùng thuộc loại hay nói nhầm "n" thành "l".

Một phương án cụ thể thực hiện theo sơ đồ khối trình bày trên Hình 2 có hai bước đầu tiên, bước 200 và bước 202, giống với phương án đã trình bày trên Hình 1, và bước 204 chỉ có khác biệt với bước 104 của phương án đã trình bày trên Hình 1 ở công đoạn cuối cùng, đó là:

nếu $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0', w_1, w_2, ..., w_b) \log_b(\alpha_{w_0'[1]}+b) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0, w_1, w_2, ..., w_b)\log_b(\alpha_{w_0[1]}+b)$thì:

sửa w_0 thành w_0';

cộng thêm 1 vào giá trị cho$\alpha_{w_0'[1]}$;

với $w_0[1]$ và $w_0'[1]$ là các chữ cái đầu của w_0 và w_0', thuộc vào một trong hai khả năng "l" hoặc "n",$\alpha_{"l"}$ và $\alpha_{"n"}$ là tham số được lưu trữ khi quy trình được áp dụng lặp đi lặp lại riêng cho một người dùng nhất định, trong đó $\alpha_{"l"}$có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "n" thành bắt đầu bằng "l"; và $\alpha_{"n"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "l" thành bắt đầu bằng "n"; còn b là một số dương nhất định, ví dụ $b = 10$.

Có thể thay hàm số $\log_b(\alpha_{"l"}+b)$ và $\log_b(\alpha_{"n"}+b)$ bằng những hàm số khác phụ thuộc vào $\alpha_{"l"}$ và $\alpha_{"n"}$ sao cho các hàm này có giá trị bằng 1 khi $\alpha_{"l"}$ và $\alpha_{"n"}$ bằng 0, và tăng dần khi $\alpha_{"l"}$ và $\alpha_{"n"}$ tăng.

Phương án vừa nêu trên ngầm định rằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "l" thành bắt đầu bằng "n", hoặc ngược lại, tác động giống nhau lên mọi lần sửa tiếp theo "l" thành "n", hoặc ngược lại. Có thể tiếp tục cải tiến phương án này, khi nhận xét rằng thói quen dùng từ nhầm chữ cái bắt đầu "l" sang "n" ở một số người không nhất thiết là đồng đều cho mọi từ có chữ cái bắt đầu bằng "l" - tương tự với thói quen dùng từ nhầm chữ cái bắt đầu "n" sang "l" - mà có thể thường bị nhầm hơn với một số từ nhất định. Như vậy, một phương án cải tiến sẽ cần đảm bảo ghi nhận số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "l" cho riêng rẽ từng chữ.

Một phương án cụ thể để thực hiện cải tiến vừa nêu, theo sơ đồ khối trình bày trên Hình 2 có hai bước đầu tiên, bước 200 và bước 202, giống với phương án đã trình bày trên Hình 1, và bước 204 chỉ có khác biệt với bước 104 của phương án đã trình bày trên Hình 1 ở công đoạn cuối cùng, đó là:

nếu $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0' , w_1, w_2, ..., w_b)\, f(w_0'[1], \alpha_{w_0'}) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)\, f(w_0[1], \alpha_{w_0})$ thì:

sửa w_0 thành w_0';

cộng thêm 1 vào giá trị cho $\alpha_{w_0'}$;

với $w_0[1]$ và $w_0'[1]$ là các chữ cái đầu của w_0 và w_0', thuộc vào một trong hai khả năng "l" hoặc "n"; α_w, với w được dùng để chỉ w_0 hoặc w_0', là tham số được lưu trữ khi quy trình được áp dụng lặp đi lặp lại riêng cho một người

dùng nhất định, có giá trị bằng số lần mà quy trình đã tự động sửa chữ khác thành chữ w; còn $f("l", \alpha_w)$ và $f("n", \alpha_w)$là các hàm số có giá trị bằng 1 khi α_w bằng 0, và tăng dần khi α_w tăng.

Một phương án tổng quát hơn nữa để thực hiện cải tiến tương tự, theo sơ đồ khối trình bày trên Hình 2 có hai bước đầu tiên, bước 200 và bước 202, giống với phương án đã trình bày trên Hình 1, và bước 204 chỉ có khác biệt với bước 104 của phương án đã trình bày trên Hình 1 ở công đoạn cuối cùng, đó là:

nếu $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0', w_1, w_2, ..., w_b) f(w_0'[1], \alpha_{w_0'[1]}, \alpha_{w_0'}) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0, w_1, w_2, ..., w_b) f(w_0[1], \alpha_{w_0[1]}, \alpha_{w_0})$ thì:

sửa w_0 thành w_0';

cộng thêm 1 vào giá trị cho $\alpha_{w_0'}$;

cộng thêm 1 vào giá trị cho $\alpha_{w_0'[1]}$;

với $w_0[1]$ và $w_0'[1]$ là các chữ cái đầu của w_0 và w_0', thuộc vào một trong hai khả năng "l" hoặc "n"; $\alpha_{"l"}$,$\alpha_{"n"}$ và α_w, với w được dùng để chỉ w_0 hoặc w_0', là các tham số được lưu trữ khi quy trình được áp dụng lặp đi lặp lại riêng cho một người dùng nhất định, trong đó $\alpha_{"l"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "n" thành bắt đầu bằng "l", $\alpha_{"n"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "l" thành bắt đầu bằng "n", α_w có giá trị bằng số lần mà quy trình đã tự động sửa chữ khác thành chữ w; còn $f("l", \alpha_{"l"}, \alpha_w)$và $f("n", \alpha_{"n"}, \alpha_w)$ là các hàm số có giá trị bằng 1 khi $\alpha_{"l"}$,$\alpha_{"n"}$ và α_w bằng 0, và tăng dầnkhi $\alpha_{"l"}$,$\alpha_{"n"}$ và α_w tăng.

Trong quy trình trên, ví dụ cho hàm f có thể là$f("l",x,y) = f("n",x,y) = \log_{b_1}(x+b_1)\log_{b_2}(y+b_2)$ với x và y là hai biến số của hàm, b_1 và b_2 là hai số dương; chẳng hạn b_1=100, b_2=10.

Ví dụ khác cho hàm f có thể là $f("l",\alpha_{"l"}, \alpha_{nl,w}) = f("n",\alpha_{"n"}, \alpha_w) = (\alpha_w + b_w)/b_w$ với b_w có thể là số lần xuất hiện chuỗi các chữ $w_{-a}, w_{-a+1}, ..., w_{-1}, w, w_1, w_2, ..., w_b$ trong cùng dữ liệu dùng để xây dựng bảng B.

Các phương án khác

Các nội dung ở trên đã trình bày các phương án cơ bản và phương án có sử dụng thông tin về thói quen của người dùng của quy trình đề xuất bởi sáng chế. Trong các phương án trên, một số tình huống có thể gây ra khó khăn trong xác định cách viết chữ cái đầu "l" hay "n" cho từ cần sửa.

Thứ nhất, mặc dù chữ đang cần xem xét sửa có thể là w_0, các chữ xung quanh nó, như w_{-1} hay w_1, cũng có thể đang chứa lỗi nhầm lẫn chữ cái đầu "n" thành "l" hoặc ngược lại. Thực tế, khi lần lượt sửa lỗi từ đầu câu đến cuối câu đầu vào, chữ đứng sau có khả năng chưa được sửa và có khả năng chứa lỗi cao hơn.

Như vậy, thay vì chỉ sửa w_0, một giải pháp phù hợp hơn để giải quyết tình huống này là sửa cùng lúc cả w_0 và tất cả các chữ xung quanh w_0 có nguy cơ chứa lỗi nhầm lẫn chữ cái đầu "n" thành "l" hoặc ngược lại.

Cụ thể, phương án tổng quát sửa cùng lúc cả w_0 và tất cả các chữ xung quanh w_0 có nguy cơ chứa lỗi nhầm lẫn chữ cái đầu "n" thành "l" hoặc ngược lại, theo sơ đồ khối trình bày trên Hình 2 có hai bước đầu tiên, bước 200 và bước 202, giống với phương án đã trình bày trên Hình 1, và bước 204 chỉ có khác biệt với bước 104 của phương án đã trình bày trên Hình 1 ở ba công đoạn cuối cùng, đó là:

> tạo ra tập hợp T, chứa tất cả các chuỗi được sinh từ chuỗi w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b bằng cách thay thế một hoặc một số các chữ trong chuỗi này, thỏa mãn điều kiện kiểm tra ở bước thứ 2 của quy trình, bằng chữ có chữ cái đầu khác đi nhưng vẫn là "l" hoặc "n";
>
> tra từ bảng B ra các giá trị $P(t)$;
>
> nếu có chuỗi t thuộc T, thỏa mãn $P(t)\, f(\alpha_{"l"}, \alpha_{"n"}, \alpha_{t_1}, \alpha_{t_2}, ..., \alpha_{t_{a+b+1}}) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)\, f(\alpha_{"l"}, \alpha_{"n"}, \alpha_{w_{-a}}, ..., \alpha_{w_0}, ..., \alpha_{w_b})$, thì:
>
> > sửa toàn bộ chuỗi w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b thành t;
> >
> > tăng giá trị $\alpha_{"l"}$thêm một giá trị bằng với số các chữ trong t khác với chuỗi gốc w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b vì bị thay thế chữ cái đầu từ "l" thành "n";

tăng giá trị $\alpha_{"n"}$thêm một giá trị bằng với số các chữ trong t khác với chuỗi gốc w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b vì bị thay thế chữ cái đầu từ "n" thành "l";

tăng giá trị α_{t_i} thêm 1, cho mọit_i, i thuộc 1, 2, ... $a+b+1$, khác với chữ tương ứng trong chuỗi gốc w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b;

với t_1 là chữ thứ nhất trong chuỗi t, t_2 là chữ thứ hai trong chuỗi t, ... t_{a+b+1} là chữ thứ $a+b+1$ trong t;$\alpha_{"l"}$, $\alpha_{"n"}$ và α_w, với w được dùng để chỉ một chữ nào đó, là tham số được lưu trữ khi quy trình được áp dụng lặp đi lặp lại riêng cho một người dùng nhất định, trong đó $\alpha_{"l"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "n" thành bắt đầu bằng "l"; $\alpha_{"n"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "l" thành bắt đầu bằng "n"; α_w có giá trị bằng số lần mà quy trình đã tự động sửa thành w; còn f là hàm số có $a+b+3$ biến, có giá trị bằng 1 khi các $\alpha_{"l"}$, $\alpha_{"n"}$ và α_w bằng 0, và tăng dần khi $\alpha_{"l"}$, $\alpha_{"n"}$ và α_w tăng.

Thứ hai, dữ liệu dùng để xây dựng bảng B có thể đôi khi không chứa chuỗi w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., $w_b$và các chuỗi trong T. Việc này dẫn đến phép so sánh $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ với $P(t)$ cho các t thuộc T, không có ý nghĩa, vì so sánh 0 với 0.

Có thể giảm khả năng xảy ra tình huống này bằng cách thay vì dùng $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$, thì dùng $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0)\times P(w_0 , w_1, w_2, ..., w_b)$, với:

$P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0)$ bằng với số lần xuất hiện chuỗi các chữ w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 chia cho số lần xuất hiện chuỗi w_{-a}, w_{-a+1}, ..., w_{-1} trước một chữ nào đó, trong dữ liệu xây dựng B;

$P(w_0 , w_1, w_2, ..., w_b)$ bằng với số lần xuất hiện chuỗi các chữ w_0 , w_1, w_2, ..., w_b chia cho số lần xuất hiện chuỗi w_1, w_2, ..., w_b sau một chữ nào đó, trong dữ liệu xây dựng B;

trong phép thống kê vừa nêu, khi không có đủ a chữ đứng trước hay b chữ đứng sau một chữ nhất định trong câu ở dữ liệu thống kê, các chữ bị thiếu cũng đặt bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL".

Các $P(t)$ khác cũng được thay bằng tích của hai biểu thức theo kỹ thuật nêu trên. Kỹ thuật này đạt hiệu quả vì chuỗi ngắn, như w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 hay w_0 , w_1, w_2, ..., w_b dễ xảy ra hơn so với chuỗi dài, như w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b, trong dữ liệu dùng để xây dựng B.

Yêu cầu bảo hộ

1. Vật ghi đọc được bằng máy tính có chứa các lệnh được mã hóa để thực hiện quy trình tự động sửa lỗi văn bản tiếng Việt chứa những chữ bị ghi nhầm chữ cái đầu "l" thành "n" và ngược lại, từ một câu tiếng Việt đầu vào, quy trình này bao gồm các bước:

phân tách câu đầu vào thành chuỗi các chữ; chẳng hạn nhờ các dấu cách;

lần lượt kiểm tra các chữ, theo một thứ tự nhất định, ví dụ từ đầu tới cuối chuỗi các chữ, xem có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và sau chữ cái "n" không phải chữ cái "g" hoặc "h" không, hoặc kiểm tra xem các chữ có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và chữ cái thứ hai sau "l" hoặc "n" là nguyên âm không;

với mỗi chữ thỏa mãn điều kiện kiểm tra ở bước trên thì thực hiện:

lấy ra chữ này, gọi là w_0, và a chữ đứng trước chữ w_0 trong chuỗi các chữ, gọi là w_{-a}, w_{-a+1}, ..., w_{-1}, và b chữ đứng sau w_0 trong chuỗi các chữ, gọi là w_1, w_2, ..., w_b, với a và b là hai số nguyên đã định trước, ví dụ $a = 1$ và $b = 1$; trong trường hợp không có đủ a chữ đứng trước w_0 hoặc không có đủ b chữ đứng sau w_0, thì bỏ qua các chữ còn thiếu hoặc các chữ còn thiếu được mặc định bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

đưa w_0 và w_{-a}, w_{-a+1}, ..., w_{-1} và w_1, w_2, ..., w_b vào một bảng tra đã lập sẵn để tra ra một con số tương ứng với bộ các chữ cho vào, gọi là $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$; bảng tra này gọi là bảng B; chẳng hạn được xây dựng bằng cách thống kê lại từ một lượng lớn các câu tiếng Việt, trong đó:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với số lần xuất hiện chuỗi các chữ w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b chia cho số lần xuất hiện chuỗi w_{-a}, w_{-a+1}, ..., w_{-1} trước một chữ nào đó rồi tiếp đến là chuỗi w_1, w_2, ..., w_b;

hoặc:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với:

số lần xuất hiện chuỗi các chữ $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0$ chia cho

số lần xuất hiện chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}$ trước một chữ nào đó nhân với

số lần xuất hiện chuỗi các chữ $w_0, w_1, w_2, ..., w_b$ chia cho

số lần xuất hiện chuỗi $w_1, w_2, ..., w_b$ sau một chữ nào đó;

trong phép thống kê để xây dựng bảng B như vừa nêu, khi không có đủ a chữ đứng trước hay b chữ đứng sau một chữ nhất định trong câu ở dữ liệu thống kê, các chữ bị thiếu cũng được bỏ qua hoặc đặt bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

tạo ra chữ mới w_0', là chữ w_0 có chữ cái đầu khác đi nhưng vẫn là "l" hoặc "n";

tra từ bảng B ra $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0' , w_1, w_2, ..., w_b)$;

sửa w_0 thành w_0' nếu $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0' , w_1, w_2, ..., w_b) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$.

2. Vật ghi đọc được bằng máy tính có chứa các lệnh được mã hóa để thực hiện quy trình tự động sửa lỗi văn bản tiếng Việt chứa những chữ ghi nhầm chữ cái đầu "l" thành "n" và ngược lại, từ một câu tiếng Việt đầu vào, quy trình bao gồm các bước:

phân tách câu đầu vào thành chuỗi các chữ; chẳng hạn nhờ các dấu cách;

lần lượt kiểm tra các chữ, theo một thứ tự nhất định, ví dụ từ đầu tới cuối chuỗi các chữ, xem có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và sau chữ cái "n" không phải chữ cái "g" hoặc "h" không, hoặc kiểm tra xem các chữ có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và chữ cái thứ hai sau "l" hoặc "n" là nguyên âm không;

với mỗi chữ thỏa mãn điều kiện kiểm tra ở bước trên thì thực hiện:

lấy ra chữ này, gọi là w_0, và a chữ đứng trước chữ w_0 trong chuỗi các chữ, gọi là w_{-a}, w_{-a+1}, ..., w_{-1}, và b chữ đứng sau w_0 trong chuỗi các chữ, gọi là w_1, w_2, ..., w_b, với a và b là hai số nguyên đã định trước, ví dụ $a = 1$ và $b = 1$; trong trường hợp không có đủ a chữ đứng trước w_0 hoặc không có đủ b chữ đứng sau w_0, thì bỏ qua các chữ còn thiếu hoặc các chữ còn thiếu được mặc định bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

đưa w_0 và w_{-a}, w_{-a+1}, ..., w_{-1} và w_1, w_2, ..., w_b vào một bảng tra đã lập sẵn để tra ra một con số tương ứng với bộ các chữ cho vào, gọi là $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$; bảng tra này gọi là bảng B; chẳng hạn được xây dựng bằng cách thống kê lại từ một lượng lớn các câu tiếng Việt, trong đó:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với số lần xuất hiện chuỗi các chữ w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b chia cho số lần xuất hiện chuỗi w_{-a}, w_{-a+1}, ..., w_{-1} trước một chữ nào đó rồi tiếp đến là chuỗi w_1, w_2, ..., w_b;

hoặc:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với:

số lần xuất hiện chuỗi các chữ w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 chia cho

số lần xuất hiện chuỗi w_{-a}, w_{-a+1}, ..., w_{-1} trước một chữ nào đó nhân với

số lần xuất hiện chuỗi các chữ w_0, w_1, w_2, ..., w_b chia cho

số lần xuất hiện chuỗi w_1, w_2, ..., w_b sau một chữ nào đó;

trong phép thống kê để xây dựng bảng B như vừa nêu, khi không có đủ a chữ đứng trước hay b chữ đứng sau một chữ nhất định trong câu ở dữ liệu thống kê, các chữ bị thiếu cũng được bỏ qua hoặc đặt bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

tạo ra chữ mới w_0', là chữ w_0 có chữ cái đầu khác đi nhưng vẫn là "l" hoặc "n";

tra từ bảng B ra $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0', w_1, w_2, ..., w_b)$;

nếu $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0', w_1, w_2, ..., w_b) f(w_0'[1], \alpha_{w_0'[1]}, \alpha_{w_0'}) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0, w_1, w_2, ..., w_b) f(w_0[1], \alpha_{w_0[1]}, \alpha_{w_0})$ thì:

sửa w_0 thành w_0';

cộng thêm 1 vào giá trị cho $\alpha_{w_0'}$;

cộng thêm 1 vào giá trị cho $\alpha_{w_0'[1]}$;

với $w_0[1]$ và $w_0'[1]$ là các chữ cái đầu của w_0 và w_0', thuộc vào một trong hai khả năng "l" hoặc "n"; $\alpha_{"l"}$,$\alpha_{"n"}$ và α_w, với w được dùng để chỉ w_0 hoặc w_0', là các tham số được lưu trữ khi quy trình được áp dụng lặp đi lặp lại riêng cho một người dùng nhất định, trong đó $\alpha_{"l"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "n" thành bắt đầu bằng "l", $\alpha_{"n"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "l" thành bắt đầu bằng "n", α_w có giá trị bằng số lần mà quy trình đã tự động sửa chữ khác thành chữ w; còn f("l", $\alpha_{"l"}$, α_w) và f("n", $\alpha_{"n"}$, α_w) là các hàm số có giá trị bằng 1 khi $\alpha_{"l"}$,$\alpha_{"n"}$ và α_w bằng 0, và không giảm khi $\alpha_{"l"}$,$\alpha_{"n"}$ và α_w tăng.

3. Vật ghi đọc được bằng máy tính có chứa các lệnh được mã hóa để thực hiện quy trình tự động sửa lỗi văn bản tiếng Việt chứa những chữ ghi nhầm chữ cái đầu "l" thành "n" và ngược lại, từ một câu tiếng Việt đầu vào, quy trình bao gồm các bước:

phân tách câu đầu vào thành chuỗi các chữ; chẳng hạn nhờ các dấu cách;

lần lượt kiểm tra các chữ, theo một thứ tự nhất định, ví dụ từ đầu tới cuối chuỗi các chữ, xem có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và sau chữ cái "n" không phải chữ cái "g" hoặc "h" không, hoặc kiểm tra xem các chữ có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và chữ cái thứ hai sau "l" hoặc "n" là nguyên âm không;

với mỗi chữ thỏa mãn điều kiện kiểm tra ở bước trên thì thực hiện:

lấy ra chữ này, gọi là w_0, và a chữ đứng trước chữ w_0 trong chuỗi các chữ, gọi là w_{-a}, w_{-a+1}, ..., w_{-1}, và b chữ đứng sau w_0 trong chuỗi các chữ, gọi là w_1, w_2, ..., w_b, với a và b là hai số nguyên đã định trước, ví dụ $a = 1$ và $b = 1$; trong trường hợp không có đủ a chữ đứng trước w_0 hoặc không có đủ b chữ đứng sau w_0, thì bỏ qua các chữ còn thiếu hoặc các chữ còn thiếu được mặc định bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

đưa w_0 và w_{-a}, w_{-a+1}, ..., w_{-1} và w_1, w_2, ..., w_b vào một bảng tra đã lập sẵn để tra ra một con số tương ứng với bộ các chữ cho vào, gọi là $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$; bảng tra này gọi là bảng B; chẳng hạn được xây dựng bằng cách thống kê lại từ một lượng lớn các câu tiếng Việt, trong đó:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với số lần xuất hiện chuỗi các chữ w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b chia cho số lần xuất hiện chuỗi w_{-a}, w_{-a+1}, ..., w_{-1} trước một chữ nào đó rồi tiếp đến là chuỗi w_1, w_2, ..., w_b;

hoặc:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với:

số lần xuất hiện chuỗi các chữ w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 chia cho

số lần xuất hiện chuỗi w_{-a}, w_{-a+1}, ..., w_{-1} trước một chữ nào đó nhân với

số lần xuất hiện chuỗi các chữ w_0, w_1, w_2, ..., w_b chia cho

số lần xuất hiện chuỗi w_1, w_2, ..., w_b sau một chữ nào đó;

trong phép thống kê để xây dựng bảng B như vừa nêu, khi không có đủ a chữ đứng trước hay b chữ đứng sau một chữ nhất định trong câu ở dữ liệu thống kê, các chữ bị thiếu cũng được bỏ qua hoặc đặt bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

tạo ra tập hợp T, chứa tất cả các chuỗi được sinh từ chuỗi w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b bằng cách thay thế một hoặc một số các chữ trong chuỗi này, thỏa mãn điều kiện kiểm tra ở bước thứ 2 của quy trình, bằng chữ có chữ cái đầu khác đi nhưng vẫn là "l" hoặc "n";

tra từ bảng B ra các giá trị $P(t)$;

nếu có chuỗi t thuộc T, thỏa mãn $P(t) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ thì sửa toàn bộ chuỗi w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b thành t.

4. Vật ghi đọc được bằng máy tính có chứa các lệnh được mã hóa để thực hiện quy trình tự động sửa lỗi văn bản tiếng Việt chứa những chữ ghi nhầm chữ cái đầu "l" thành "n" và ngược lại, từ một câu tiếng Việt đầu vào, quy trình bao gồm các bước:

phân tách câu đầu vào thành chuỗi các chữ; chẳng hạn nhờ các dấu cách;

lần lượt kiểm tra các chữ, theo một thứ tự nhất định, ví dụ từ đầu tới cuối chuỗi các chữ, xem có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và sau chữ cái "n" không phải chữ cái "g" hoặc "h" không, hoặc kiểm tra xem các chữ có bắt đầu bằng chữ cái "l" hoặc bắt đầu bằng chữ cái "n" và chữ cái thứ hai sau "l" hoặc "n" là nguyên âm không;

với mỗi chữ thỏa mãn điều kiện kiểm tra ở bước trên thì thực hiện:

lấy ra chữ này, gọi là w_0, và a chữ đứng trước chữ w_0 trong chuỗi các chữ, gọi là w_{-a}, w_{-a+1}, ..., w_{-1}, và b chữ đứng sau w_0 trong chuỗi các chữ, gọi là w_1, w_2, ..., w_b, với a và b là hai số nguyên đã định trước, ví dụ $a = 1$ và $b = 1$; trong trường hợp không có đủ a chữ đứng trước w_0 hoặc không có đủ b chữ đứng sau w_0, thì bỏ qua các chữ còn thiếu hoặc các chữ còn thiếu được mặc định bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

đưa w_0 và w_{-a}, w_{-a+1}, ..., w_{-1} và w_1, w_2, ..., w_b vào một bảng tra đã lập sẵn để tra ra một con số tương ứng với bộ các chữ cho vào, gọi là $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$; bảng tra này gọi là bảng B; chẳng hạn được xây dựng bằng cách thống kê lại từ một lượng lớn các câu tiếng Việt, trong đó:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với số lần xuất hiện chuỗi các chữ $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b$ chia cho số lần xuất hiện chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}$ trước một chữ nào đó rồi tiếp đến là chuỗi $w_1, w_2, ..., w_b$;

hoặc:

số $P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)$ bằng với:

số lần xuất hiện chuỗi các chữ $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0$ chia cho

số lần xuất hiện chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}$ trước một chữ nào đó nhân với

số lần xuất hiện chuỗi các chữ $w_0, w_1, w_2, ..., w_b$ chia cho

số lần xuất hiện chuỗi $w_1, w_2, ..., w_b$ sau một chữ nào đó;

trong phép thống kê để xây dựng bảng B như vừa nêu, khi không có đủ a chữ đứng trước hay b chữ đứng sau một chữ nhất định trong câu ở dữ liệu thống kê, các chữ bị thiếu cũng được bỏ qua hoặc đặt bằng giá trị rỗng, có thể được ký hiệu đặc biệt, chẳng hạn "NULL";

tạo ra tập hợp T, chứa tất cả các chuỗi được sinh từ chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b$ bằng cách thay thế một hoặc một số các chữ trong chuỗi này, thỏa mãn điều kiện kiểm tra ở bước thứ 2 của quy trình, bằng chữ có chữ cái đầu khác đi nhưng vẫn là "l" hoặc "n";

tra từ bảng B ra các giá trị $P(t)$;

nếu có chuỗi t thuộc T, thỏa mãn $P(t)\, f(\alpha_{"l"}, \alpha_{"n"}, \alpha_{t_1}, \alpha_{t_2}, ..., \alpha_{t_{a+b+1}}) > P(w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b)\, f(\alpha_{"l"}, \alpha_{"n"}, \alpha_{w_{-a}}, ..., \alpha_{w_0}, ..., \alpha_{w_b})$, thì:

sửa toàn bộ chuỗi $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b$ thành t;

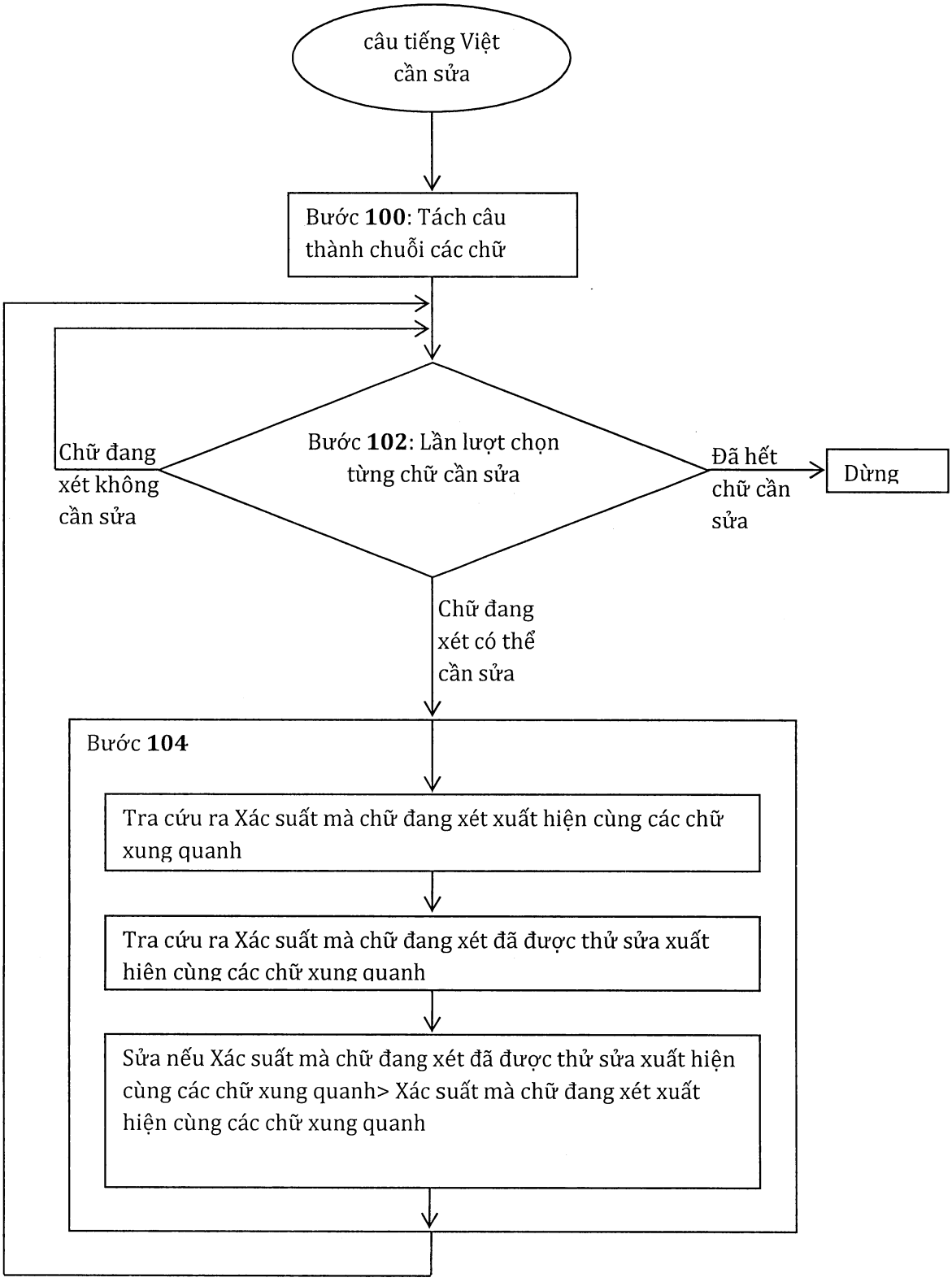
tăng giá trị $\alpha_{"l"}$ thêm một giá trị bằng với số các chữ trong t khác với chuỗi gốc $w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b$ vì bị thay thế chữ cái đầu từ "l" thành "n";

tăng giá trị $\alpha_{"n"}$ thêm một giá trị bằng với số các chữ trong t khác với chuỗi gốc w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b vì bị thay thế chữ cái đầu từ "n" thành "l";

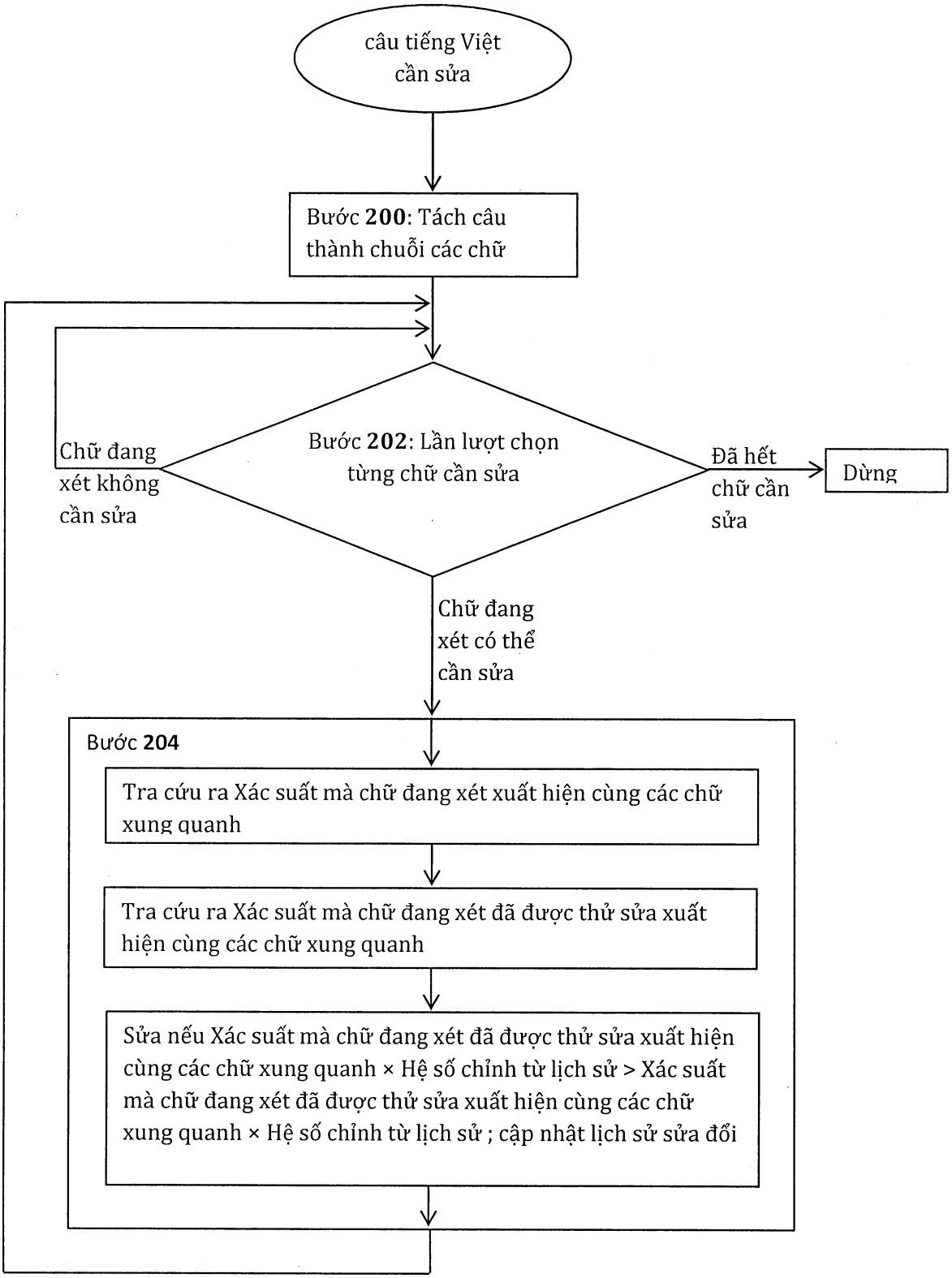
tăng giá trị α_{t_i} thêm 1, cho mọi t_i, i thuộc 1, 2, ... $a+b+1$, khác với chữ tương ứng trong chuỗi gốc w_{-a}, w_{-a+1}, ..., w_{-1}, w_0 , w_1, w_2, ..., w_b;

với t_1 là chữ thứ nhất trong chuỗi t, t_2 là chữ thứ hai trong chuỗi t, ... t_{a+b+1} là chữ thứ $a+b+1$ trong t; $\alpha_{"l"}$, $\alpha_{"n"}$ và α_w, với w được dùng để chỉ một chữ nào đó, là tham số được lưu trữ khi quy trình được áp dụng lặp đi lặp lại riêng cho một người dùng nhất định, trong đó $\alpha_{"l"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "n" thành bắt đầu bằng "l"; $\alpha_{"n"}$ có giá trị bằng số lần mà quy trình đã tự động sửa các chữ bắt đầu bằng "l" thành bắt đầu bằng "n"; α_w có giá trị bằng số lần mà quy trình đã tự động sửa thành w; còn f là hàm số có $a+b+3$ biến, có giá trị bằng 1 khi các $\alpha_{"l"}$, $\alpha_{"n"}$ và α_w bằng 0, và không giảm khi $\alpha_{"l"}$, $\alpha_{"n"}$ và α_w tăng.

Hình 1


câu tiếng Việt cần sửa
Bước **100**: Tách câu thành chuỗi các chữ
Bước **102**: Lần lượt chọn từng chữ cần sửa
Chữ đang xét không cần sửa
Đã hết chữ cần sửa
Dừng
Chữ đang xét có thể cần sửa
Bước **104**
Tra cứu ra Xác suất mà chữ đang xét xuất hiện cùng các chữ xung quanh
Tra cứu ra Xác suất mà chữ đang xét đã được thử sửa xuất hiện cùng các chữ xung quanh
Sửa nếu Xác suất mà chữ đang xét đã được thử sửa xuất hiện cùng các chữ xung quanh> Xác suất mà chữ đang xét xuất hiện cùng các chữ xung quanh

Hình 2


câu tiếng Việt cần sửa
Bước 200: Tách câu thành chuỗi các chữ
Bước 202: Lần lượt chọn từng chữ cần sửa
Chữ đang xét không cần sửa
Đã hết chữ cần sửa
Dừng
Chữ đang xét có thể cần sửa
Bước 204
Tra cứu ra Xác suất mà chữ đang xét xuất hiện cùng các chữ xung quanh
Tra cứu ra Xác suất mà chữ đang xét đã được thử sửa xuất hiện cùng các chữ xung quanh
Sửa nếu Xác suất mà chữ đang xét đã được thử sửa xuất hiện cùng các chữ xung quanh × Hệ số chỉnh từ lịch sử > Xác suất mà chữ đang xét đã được thử sửa xuất hiện cùng các chữ xung quanh × Hệ số chỉnh từ lịch sử ; cập nhật lịch sử sửa đổi