(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) **1-0048667**

(51)[2022.01] **G06F 16/00**

(13) **B**

(21) 1-2022-04119

(22) 30/06/2022

(45) 25/07/2025 448

(43) 27/03/2023 420A

(73) TẬP ĐOÀN CÔNG NGHIỆP - VIỄN THÔNG QUÂN ĐỘI (VN)
Lô D26 Khu đô thị mới Cầu Giấy, phường Yên Hoà, quận Cầu Giấy, thành phố Hà Nội

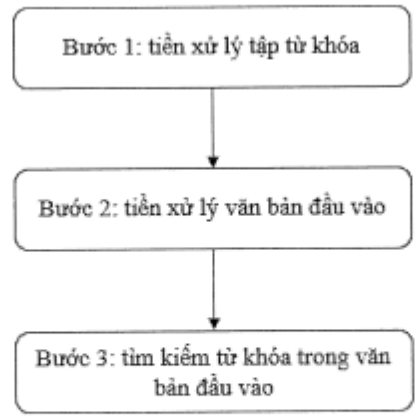
(72) Nguyễn Thị Hoài Phương (VN); Nguyễn Xuân Tiến (VN); Chu Hữu Tiến Nam (VN).

(74) Công ty TNHH NACILAW (NACILAW)

(54) PHƯƠNG PHÁP TÌM KIẾM TẬP TỪ KHÓA TRONG VĂN BẢN

(21) 1-2022-04119

(57) Phương pháp tìm kiếm tập từ khóa trong văn bản bao gồm các bước: bước 1: tiền xử lý tập từ khóa; bước 2: tiền xử lý văn bản đầu vào; bước 3: tìm kiếm từ khóa trong văn bản đầu vào. Đánh giá chung, trong các trường hợp, phương pháp tìm kiếm tập từ khóa trong văn bản này đảm bảo được các yếu tố: tối ưu về cả tốc độ tìm kiếm lẫn số lượng tập từ khóa tìm kiếm, đồng thời có thể linh động thêm bớt từ khóa trong quá trình tìm kiếm mà không ảnh hưởng hiệu năng.



Hình 3

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp tìm kiếm tập từ khóa trong văn bản. Cụ thể, phương pháp tìm kiếm tập từ khóa trong văn bản được đề cập trong sáng chế có thể áp dụng cho các sản phẩm phần mềm có nghiệp vụ xử lý văn bản như tìm kiếm nội dung, trích xuất nội dung, so sánh độ tương đồng của các văn bản, kiểm tra chính tả, lọc thư rác, lọc tin nhắn rác, máy tìm kiếm.

Tình trạng kỹ thuật của sáng chế

Các phương pháp xử lý vấn đề tìm kiếm từ khóa trong văn bản gồm có các phương pháp nổi bật sau: phương pháp Naïve, phương pháp Rabin - Karp, phương pháp Knuth - Morris - Pratt (KMP), phương pháp Aho - Corasick …Trong đó các phương pháp Naïve, KMP chỉ áp dụng được cho việc tìm kiếm đơn từ khóa (chỉ tìm duy nhất một từ khóa trong văn bản), không áp dụng được cho việc tìm kiếm đa từ khóa. Các phương pháp Rabin - Karp, Aho - Corasick được dùng cho việc tìm kiếm đa từ khóa. Cụ thể:

Phương pháp Naïve: đây là cách đơn giản và không hiệu quả nhất để tìm kiếm từ khóa trong văn bản. Cách làm của phương pháp này là duyệt lần lượt từng ký tự trong nội dung văn bản, với mỗi ký tự sẽ so sánh với ký tự đầu tiên của từ khóa, nếu so sánh khớp thì tiếp tục so sánh ký tự tiếp theo của chuỗi với ký tự tiếp theo của từ khóa, tìm kiếm là thành công nếu so sánh khớp liên tiếp đến ký tự cuối cùng của từ khóa. Nếu so sánh không khớp ở bước bất kỳ thì lại lặp lại so sánh ký tự tiếp theo trong vòng lặp với ký tự đầu tiên của từ khóa. Trong trường hợp trung bình, độ phức tạp của phương pháp này là O (n + m), còn trường hợp tồi nhất thì độ phức tạp là O (n*m), trong đó n là độ dài của văn bản và m là độ dài của từ khóa. Phương pháp này chỉ áp dụng được cho vấn đề tím kiếm đơn từ khóa, vì độ phức tạp khi tìm kiếm đa từ khóa là rất lớn và phụ thuộc vào số lượng từ khóa. Trường hợp tìm kiếm cho k từ khóa thì độ phức tạp trung bình của phương pháp này là O (k*(n+m)), và trong trường hợp tồi nhất là O (k*n*m).

Phương pháp Rabin - Karp: so với phương pháp Naïve, phương pháp Rabin - Karp sử dụng hàm băm để tăng tốc độ tìm kiếm từ khóa. Thay vì tìm kiếm lần lượt tất cả các vị trí trong văn bản, phương pháp Rabin - Karp sẽ kiểm tra giá trị băm tại mỗi vị trí văn bản, và chỉ thực hiện tìm kiếm tại vị trí có giá trị băm bằng với giá trị băm của từ khóa, việc này giúp giảm số lần phải tìm kiếm. Giả sử hàm băm tốt, độ phức tạp tính toán hàm băm là O

(1), thì giải thuật Rabin - Karp khi tìm kiếm đa từ khóa có độ phức tạp O (n + km), tương đương với giải thuật Aho - Corasick.

Phương pháp KMP: phương pháp KMP có độ phức tạp trong trường hợp tồi nhất tốt hơn so với phương pháp Naïve. KMP dành thời gian để tiền xử lý từ khóa, lưu lại kết quả trong bảng tiền tố, từ đó làm tăng tốc độ tìm kiếm hơn so với Naïve. Nhờ có bảng tiền tố, việc so khớp nội dung văn bản không phải thực hiện tìm kiếm với tất cả từng vị trí trong văn bản, mà chỉ cần tìm kiếm tại những vị trí dự phòng được lưu vết trong bảng tiền tố. Về độ phức tạp tính toán, thời gian xử lý hàm tiền tố là O (m) và thời gian xử lý tìm kiếm là O (n), tổng thời gian xử lý cho phương pháp KMP là O (n + m), trong đó n là độ dài của chuỗi văn bản, m là độ dài của từ khóa. KMP chỉ mô tả bài toán tìm kiếm đơn từ khóa.

Phương pháp Aho - Corasick: phương pháp này được coi là mở rộng của phương pháp KMP, vì nó cũng áp dụng tư tưởng về hàm tiền tố của KMP để xử lý bài toán tìm kiếm đa từ khóa. Đây được đánh giá là một trong những phương pháp hiệu quả trong bài toán tìm kiếm từ khóa. Độ phức tạp của phương pháp Aho - Corasick là O (m) cho phần tiền xử lý tập từ khóa xây dựng cây đồ thị, và O (n + z) cho phần tìm kiếm, O (m + n + z) cho tổng cả quá trình xử lý phương pháp, trong đó n là độ dài của chuỗi văn bản, m là tổng độ dài của tập từ khóa và z là tổng độ dài của những từ khóa theo số lần tìm kiếm thành công. Phương pháp Aho – Corasick giả định rằng tập từ khóa tìm kiếm là cố định, tức không thay đổi (thêm bớt linh động) trong quá trình tìm kiếm, bởi việc thêm bớt chỉ 1 từ khóa làm thay đổi cây đồ thị với độ phức tạp O (m), nên khi đó việc tìm kiếm từ khóa sẽ tăng thời gian xử lý từ O (n + z) lên thành O (m + n + z). Khi tập từ khóa rất lớn (không giới hạn) thì m là rất lớn, đây trở thành điểm yếu của phương pháp này.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là thực hiện tìm kiếm sự xuất hiện của tập từ khóa trong văn bản khắc phục được các hạn chế của các phương pháp đã biết. Giá trị cốt lõi của phương pháp này nằm ở tính hiệu quả. Phương pháp cho lời giải chính xác theo thời gian thực, không phụ thuộc vào số lượng từ khóa, không phụ thuộc vào kiểu mã hóa của từ khóa cũng như văn bản. Số lượng từ khóa là không giới hạn và việc khai báo động từ khóa không ảnh hưởng đến hiệu năng cũng như tính chính xác của việc tìm kiếm từ khóa trong nội dung. Phương pháp này chạy đúng cho mọi định dạng mã hóa (tiếng Anh, tiếng Việt, tiếng Myanmar, tiếng Tây Ban Nha, chuỗi nhị phân bất kỳ …).

Để đạt được mục đích trên, phương pháp này bao gồm ba bước cơ bản:

Bước 1: tiền xử lý tập từ khóa. Tại bước này, từng từ khóa sẽ được phân tách bởi các ký tự ngăn cách theo danh sách có thể cấu hình thành hai phần: phần đầu và phần đuôi. Từ khoá sau khi được phân tách sẽ lưu vào bảng dữ liệu gồm hai trường (trường "từ khóa" chứa phần đầu của từ khóa sau khi phân tách, và trường "nội dung" chứa toàn bộ từ khóa).

Bước 2: tiền xử lý văn bản đầu vào. Tại bước này, phần văn bản được tiền xử lý sẽ chia thành hai phần: phần đầu và phần đuôi được phân tách bởi các ký tự ngăn cách theo danh sách cấu hình, và được lưu lại trong danh sách thành hai phần: phần từ khóa chính là phần đầu phần văn bản được xử lý và phần nội dung chính là toàn bộ phần văn bản được xử lý.

Bước 3: tìm kiếm sự xuất hiện của từ khóa trong văn bản đầu vào. Tại bước này, thực hiện tìm kiếm trong bảng lưu trữ tập từ khóa theo phần từ khóa trong mỗi cụm văn bản.

Mô tả vắn tắt các hình vẽ

Hình 1: hình minh họa phương pháp của sáng chế, bước tìm kiếm trong bảng lưu trữ từ khóa với văn bản có nội dung *"ngày hội giảm-giá siêu*khuyến mại"* và tập từ khóa *("khuyến mại", "giảm-giá", "giảm_sâu", "siêu*khuyến*mại")*;

Hình 2: hình minh họa phương pháp của sáng chế, bước so sánh nội dung trong văn bản tìm tiền tố chứa từ khóa với văn bản có nội dung *"ngày hội giảm-giá siêu*khuyến mại"* và tập từ khóa *("khuyến mại", "giảm-giá", "giảm_sâu", "siêu*khuyến*mại")*;

Hình 3: là hình minh họa các bước được thực hiện trong sáng chế.

Mô tả chi tiết sáng chế

Tham chiếu Hình 3, phương pháp tìm kiếm tập từ khóa trong văn bản gồm ba phần xử lý chính như sau:

Bước 1: tiền xử lý tập từ khóa.

Tại bước này, từng từ khóa sẽ được phân tách thành hai phần: phần đầu và phần đuôi. Các từ khóa được phân tách bởi các ký tự ngăn cách theo danh sách có thể cấu hình (ví dụ các ký tự ngăn cách như dấu cách, dấu phẩy, dấu chấm…). Thực hiện duyệt từng từ khóa, mỗi từ khóa duyệt từng ký tự từ trái qua phải. Nếu ký tự đang duyệt không là ký tự ngăn cách thì tiếp tục duyệt tiếp sang ký tự tiếp theo của từ khóa, nếu ký tự đang duyệt là ký tự ngăn cách thì dừng duyệt, cập nhật từ khóa vào bảng từ khóa (với phần đầu của từ khóa là chuỗi ký tự tính từ ký tự đầu tiên đến ký tự ngăn cách đầu tiên vừa tìm được), và

duyệt sang từ khóa tiếp theo. Từ khóa sau khi được phân tách sẽ lưu vào bảng dữ liệu cấu trúc như sau:

Từ khóa	Nội dung
Phần đầu của từ khóa	Toàn bộ từ khóa

Bảng 1: cấu trúc dữ liệu bảng từ khóa

Ví dụ với tập từ khóa *"khuyến mại"*, *"giảm-giá"*, *"giảm_sâu"*, *"siêu*khuyến*mại"*, và tập các ký tự ngăn cách gồm *" "* (dấu cách), *"-"* (dấu gạch ngang), *"*"* (dấu sao), thì bảng dữ liệu từ khóa sau khi phân tách sẽ như sau:

Từ khóa	Nội dung
khuyến	khuyến mại
giảm	giảm-giá
	giảm_sâu
Siêu	siêu*khuyến*mại

Bảng 2: ví dụ tập từ khóa sau khi xử lý

Bước 2: tiền xử lý văn bản đầu vào.

Văn bản đầu vào trước khi thực hiện tìm kiếm cũng được tách thành nhiều cụm, mỗi cụm gồm hai phần: phần đầu và phần đuôi, được phân tách bởi các ký tự ngăn cách theo danh sách cấu hình. Thực hiện vòng lặp, bắt đầu bằng vòng lặp đầu tiên với phần văn bản được xử lý chính là toàn bộ văn bản. Phần văn bản được tiền xử lý sẽ chia thành hai phần: phần đầu và phần đuôi được phân tách bởi các ký tự ngăn cách theo danh sách cấu hình, và được lưu lại trong danh sách thành hai phần: phần từ khóa chính là phần đầu phần văn bản được xử lý và phần nội dung chính là toàn bộ phần văn bản được xử lý. Chuyển sang vòng lặp tiếp theo với phần văn bản được xử lý ở vòng lặp này là phần đuôi của phần văn bản được xử lý ở vòng lặp liền trước đó. Lặp lại quá trình cho đến khi hết văn bản (tức là phần văn bản được xử lý bằng rỗng). Văn bản sau khi được tiền xử lý sẽ được lưu thành một danh sách các cụm văn bản.

Ví dụ với văn bản có nội dung *"ngày hội giảm-giá siêu*khuyến mại"*, và tập các ký tự ngăn cách gồm *" "* (dấu cách), *"-"* (dấu gạch ngang), *"*"* (dấu sao), thì văn bản sau khi phân tách sẽ như sau:

Từ khóa	Nội dung
ngày	ngày hội giảm-giá siêu*khuyến mại
hội	hội giảm-giá siêu*khuyến mại
giảm	giảm-giá siêu*khuyến mại
giá	giá siêu*khuyến mại
siêu	siêu*khuyến mại
khuyến	khuyến mại
mại	mại

Bảng 3: ví dụ văn bản sau khi xử lý

Bước 3: tìm kiếm từ khóa trong văn bản đầu vào.

Với văn bản đã qua bước tiền xử lý thành danh sách các cụm văn bản, thực hiện tìm kiếm bằng cách duyệt từng cụm văn bản, tìm kiếm trong bảng lưu trữ tập từ khóa theo phần từ khóa trong mỗi cụm văn bản.

- Trường hợp không tìm thấy bản ghi nào trong bảng lưu trữ tập từ khóa, thì tức là không tìm thấy từ khóa nào trong cụm văn bản đang xem xét.
- Trường hợp có tìm thấy bản ghi trong bảng lưu trữ tập từ khóa, thì tức là có khả năng cụm văn bản này chứa từ khóa cần tìm. Tiếp tục so sánh trường nội dung của bản ghi tìm được có là tiền tố của phần nội dung trong cụm văn bản hay không
    - Nếu trường nội dung của bản ghi là tiền tố của phần nội dung trong cụm văn bản, thì được cho là tìm thấy sự xuất hiện của từ khóa trong văn bản đầu vào.
    - Nếu trường nội dung của bản ghi không là tiền tố của phần nội dung trong cụm văn bản, thì tức là không tìm thấy từ khóa nào trong cụm văn bản đang xem xét.

Ví dụ thực hiện sáng chế

Ví dụ với văn bản có nội dung *"ngày hội giảm-giá siêu*khuyến mại"* và tập từ khóa (*"khuyến mại", "giảm-giá", "giảm_sâu", "siêu*khuyến*mại"*), việc tìm kiếm được mô phỏng như hình sau:

Bước 1: tìm kiếm trong bảng lưu trữ từ khóa theo các từ khóa trong cụm văn bản: *"ngày", "hội", "giảm", "giá", "siêu", "khuyến", "mại"*. Kết quả tìm thấy ba từ khóa sau

trong bảng lưu trữ từ khóa: *"giảm", "siêu", "khuyến"*. Với ba cụm văn bản tìm được từ khóa trong bảng lưu trữ, tiếp tục chuyển sang bước 2. Với những cụm văn bản còn lại thực hiện dừng xử lý do không tìm thấy từ khóa phù hợp. Chi tiết bước 1 như Hình 1 minh họa dưới đây.

Bước 2: so sánh nội dung văn bản tìm tiền tố chứa từ khóa. Sau khi kết thúc bước 1, tìm được ba cụm văn bản có chứa tiền tố của từ khóa đó là *"giảm-giá siêu*khuyến mại", "siêu* khuyến mại"* và *"khuyến mại"*. Đây chính là điều kiện cần của quá trình tìm kiếm. Bước 2 tiếp tục kiểm tra điều kiện đủ là xem xét nội dung ba cụm văn bản này có bắt đầu bởi toàn bộ từ khóa tìm thấy hay không (tức nội dung từ khóa có là tiền tố của cụm văn bản hay không). Kết quả kiểm tra cho thấy hai cụm văn bản *"giảm-giá siêu*khuyến mại"* và *"khuyến mại"* đạt yêu cầu tìm kiếm vì lần lượt có tiền tố là toàn bộ nội dung từ khóa *"giảm-giá"* và *"khuyến mại"*; còn cụm văn bản *"siêu*khuyến mại"* không đạt yêu cầu kiểm tra do tiền tố của cụm này chỉ chứa được một phần nội dung từ khóa (chỉ chứa *"siêu*khuyến"* chứ không chứa toàn bộ *"siêu*khuyến*mại"*). Kết quả cuối cùng là nội dung văn bản *"ngày hội giảm-giá siêu*khuyến mại"* xuất hiện hai từ khóa *"khuyến mại"* và *"giảm-giá"*. Chi tiết bước 2 như Hình 2 minh họa dưới đây.

Hiệu quả đạt được của sáng chế

Các phương pháp tìm kiếm nổi tiếng đã trình bày ở trên như Naïve, Rabin - Karp, Knuth - Morris - Pratt (KMP), Aho - Corasick đều không phù hợp với yêu cầu tìm kiếm với tập từ khóa không giới hạn số lượng, và có thể thay đổi linh động từ khóa trong quá trình tìm kiếm. Phương pháp tìm kiếm tự phát triển đã khắc phục được những nhược điểm kể trên mà vẫn đảm bảo duy trì thời gian tìm kiếm tốt tương đương với thuật toán Aho - Corasick. Cụ thể như sau:

Phần tiền xử lý tập từ khóa: với từng từ khóa thực hiện duyệt tách từ thành hai phần, việc này có độ tốn kém O (M) với M là độ dài của từ khóa, sau đó thực hiện lưu từ khóa đã xử lý tách xuống bảng dữ liệu có độ phức tạp là O (1). Như vậy tiền xử lý cả tập từ khóa sẽ có độ phức tạp là O (m), trong đó m là tổng độ dài toàn bộ tập từ khóa. Việc bớt một từ khóa sẽ cần tiền xử lý tách từ khóa và xóa thông tin từ khóa khỏi bảng dữ liệu, tương tự việc thêm một từ khóa sẽ cần tiền xử lý tách từ khóa và thêm thông tin từ khóa vào bảng dữ liệu, thêm/bớt từ khóa trong tập từ khóa ban đầu sẽ có độ phức tạp O (M), không bị ảnh hưởng thời gian bởi toàn bộ tập từ khóa, chỉ phụ thuộc vào độ dài chính từ khóa cần thêm/bớt. Đây là điểm khác biệt với thuật toán Aho - Corasick, giúp cho hệ thống có thể

thêm bớt từ khóa linh động trong quá trình tìm kiếm mà không ảnh hưởng đến hiệu năng của việc tìm kiếm, và số lượng tập từ khóa là không giới hạn.

Phần tiền xử lý văn bản đầu vào: thực hiện duyệt văn bản tách thành các cụm nhỏ, thao tác này có độ phức tạp O (n), với n là độ dài văn bản. Kết quả tách được N cụm, với N < n. Trường hợp tốt nhất, khi văn bản không có ký tự ngăn cách nào (ví dụ "abcdefghijklmnopq") thì chỉ tách được một cụm, khi đó N = 1. Trường hợp tồi nhất, khi văn bản có kiểu ký tự thường và ký tự ngăn cách xen kẽ và độ dài ký tự thường chỉ bằng 1 (ví dụ "a b c d e f g h i j k l m n o p q"), khi đó sẽ tách được N = n/2 cụm. Vậy 1 <= N <= n/2.

Phần tìm kiếm tập từ khóa trong văn bản: thực hiện so khớp từng cụm văn bản với tập từ khóa được lưu trong bảng dữ liệu bằng cách tìm kiếm theo trường khóa trong bảng dữ liệu cụm từ khóa, mỗi lần tìm kiếm là O (1), số lần tìm kiếm bằng với số cụm văn bản, độ phức tạp tìm kiếm là O (N). Khi tìm kiếm khớp phần khóa thì sẽ thực hiện so sánh toàn bộ từ khóa tìm thấy với cụm văn bản đang khớp, độ phức tạp của phần so khớp này là O (z), z là tổng độ dài của những từ khóa theo số lần tìm kiếm thành công (tương đương với định nghĩa z ở thuật toán Aho - Corasick).

Tổng kết lại, trong phương pháp tìm kiếm mới này giai đoạn tiền xử lý tập từ khóa có thời gian thực hiện O (m), phần này thực hiện từ trước và độc lập với giai đoạn tìm kiếm. Giai đoạn tìm kiếm gồm có tiền xử lý văn bản trước khi tìm kiếm và thực hiện tìm kiếm, thời gian thực hiện là O (N + z) <= O (n/2 + z), có độ phức tạp xấp xỉ bằng một nửa so với phương pháp Aho - Corasick. Như vậy phương pháp mới tối ưu hơn các phương pháp nổi tiếng đã công bố về cả tốc độ tìm kiếm lẫn số lượng tập từ khóa tìm kiếm, đồng thời có thể linh động thêm bớt từ khóa trong quá trình tìm kiếm mà không ảnh hưởng hiệu năng.

Dưới đây là bảng tóm tắt so sánh các thuật toán trên với các tiêu chí đặt ra trong yêu cầu bài toán. Trong đó:

- m là tổng độ dài tập từ khóa;
- n là độ dài văn bản tìm kiếm;
- k là số lượng từ khóa;
- z là tổng độ dài của những từ khóa theo số lần tìm kiếm thành công;
- vSearch là thuật toán tự phát triển đang trình bày trong tài liệu.

Tiêu chí	Naïve	Rabin - Karp	KMP	Aho - Corasick	vSearch
Độ phức tạp không gian	0	O (m)	O (m)	O (m)	O (m)
Độ phức tạp tiền xử lý	0	O (m)	O (m)	O (m)	O (m)
Độ phức tạp tìm kiếm	Trung bình: O (k*n + m), Tồi nhất: O (n*m)	Trung bình: O (n + m), Tồi nhất: O (n*m)	O (n)	O (n + z)	O (n/2 + z)
Độ phức tạp tìm kiếm khi đang thêm một từ khóa	Trung bình: O (k*n + m), Tồi nhất: O (n*m)	Trung bình: O (n + m), Tồi nhất: O (n*m)	Không xác định	O (n + m + z)	O (n/2 + z)
Tìm kiếm được đa từ khóa	Có	Có	Không	Có	Có
Không giới hạn số lượng từ khóa	Có	Có	Không	Có	Có
Thay đổi linh động tập từ khóa	Có	Có	Không	Không	Có

Bảng 4: so sánh các thuật toán

Yêu cầu bảo hộ

1. Phương pháp tìm kiếm tập từ khóa trong văn bản bao gồm các bước:

bước 1: tiền xử lý tập từ khóa;

tại bước này, từng từ khóa sẽ được phân tách thành hai phần: phần đầu và phần đuôi; từ khóa được phân tách bởi các ký tự ngăn cách theo danh sách có thể cấu hình, từ khóa sau khi được phân tách sẽ lưu vào bảng dữ liệu cấu trúc như sau:

từ khóa	nội dung
phần đầu của từ khóa	toàn bộ từ khóa

bước 2: tiền xử lý văn bản đầu vào;

tại bước này, bắt đầu bằng vòng lặp đầu tiên với phần văn bản được xử lý chính là toàn bộ văn bản;

phần văn bản được tiền xử lý sẽ chia thành hai phần: phần đầu và phần đuôi được phân tách bởi các ký tự ngăn cách theo danh sách cấu hình, và được lưu lại trong danh sách thành hai phần: phần từ khóa chính là phần đầu phần văn bản được xử lý và phần nội dung chính là toàn bộ phần văn bản được xử lý;

chuyển sang vòng lặp tiếp theo với phần văn bản được xử lý ở vòng lặp này là phần đuôi của phần văn bản được xử lý ở vòng lặp liền trước đó;

lặp lại quá trình cho đến khi hết văn bản (tức là phần văn bản được xử lý bằng rỗng), văn bản sau khi được tiền xử ký sẽ được lưu thành một danh sách các cụm văn bản;

bước 3: tìm kiếm từ khóa trong văn bản đầu vào;

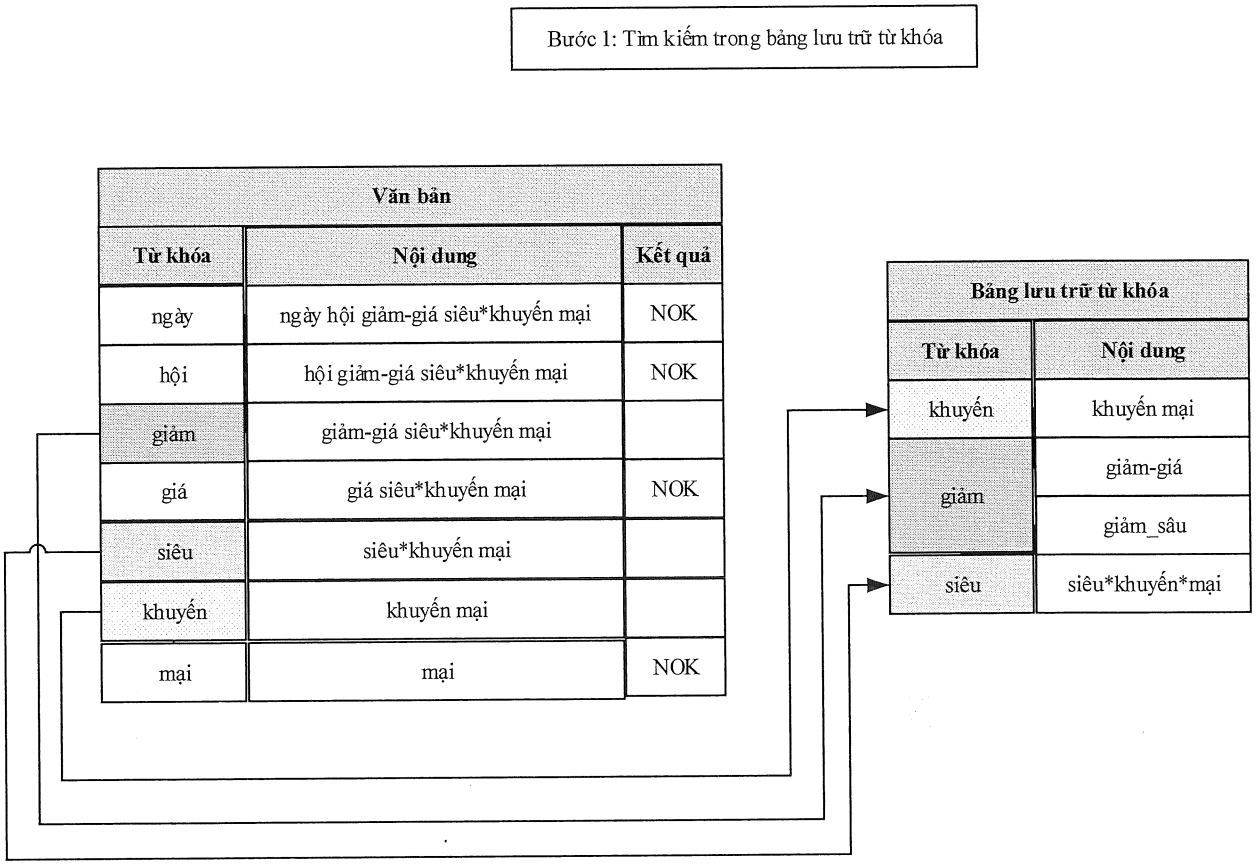
tại bước này, thực hiện tìm kiếm trong bảng lưu trữ tập từ khóa theo phần từ khóa trong mỗi cụm văn bản:

trường hợp không tìm thấy bản ghi nào trong bảng lưu trữ tập từ khóa, thì tức là không tìm thấy từ khóa nào trong cụm văn bản đang xem xét;

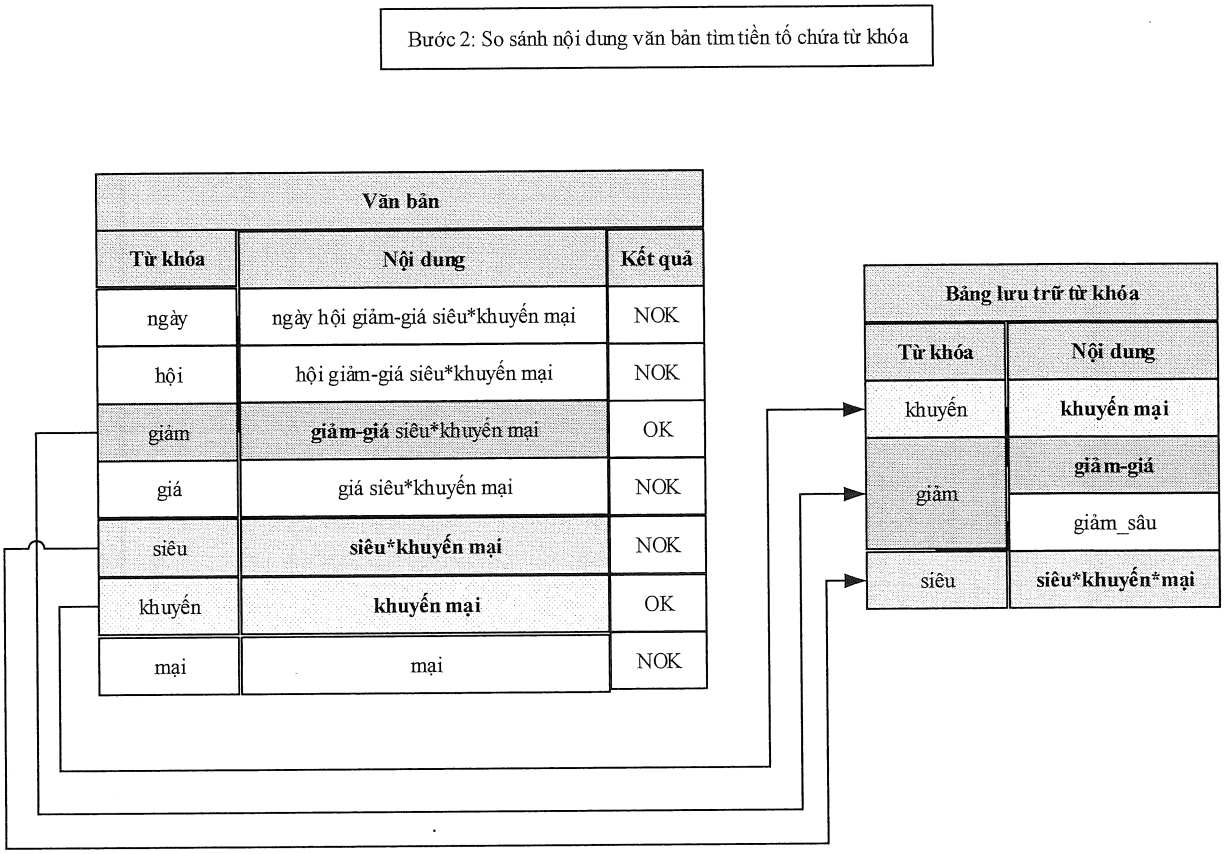
trường hợp có tìm thấy bản ghi trong bảng lưu trữ tập từ khóa, thì tức là có khả năng cụm văn bản này chứa từ khóa cần tìm, tiếp tục so sánh trường nội dung của bản ghi tìm được có là tiền tố của phần nội dung trong cụm văn bản hay không:

nếu trường nội dung của bản ghi là tiền tố của phần nội dung trong cụm văn bản, thì được cho là tìm thấy sự xuất hiện của từ khóa trong văn bản đầu vào;

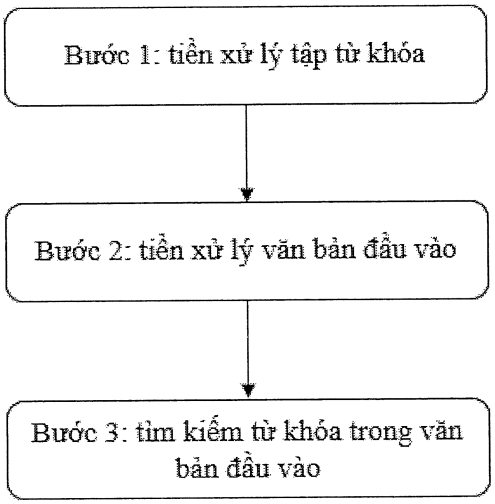
nếu trường nội dung của bản ghi không là tiền tố của phần nội dung trong cụm văn bản, thì tức là không tìm thấy từ khóa nào trong cụm văn bản đang xem xét.


Bước 1: Tìm kiếm trong bảng lưu trữ từ khóa
Văn bản
Từ khóa
Nội dung
Kết quả
ngày
ngày hội giảm-giá siêu*khuyến mại
NOK
hội
hội giảm-giá siêu*khuyến mại
NOK
giảm
giảm-giá siêu*khuyến mại
giá
giá siêu*khuyến mại
NOK
siêu
siêu*khuyến mại
khuyến
khuyến mại
mại
mại
NOK
Bảng lưu trữ từ khóa
Từ khóa
Nội dung
khuyến
khuyến mại
giảm
giảm-giá
giảm_sâu
siêu
siêu*khuyến*mại

Hình 1


Bước 2: So sánh nội dung văn bản tìm tiền tố chứa từ khóa
Văn bản
Từ khóa
Nội dung
Kết quả
ngày
ngày hội giảm-giá siêu*khuyến mại
NOK
hội
hội giảm-giá siêu*khuyến mại
NOK
giảm
giảm-giá siêu*khuyến mại
OK
giá
giá siêu*khuyến mại
NOK
siêu
siêu*khuyến mại
NOK
khuyến
khuyến mại
OK
mại
mại
NOK
Bảng lưu trữ từ khóa
Từ khóa
Nội dung
khuyến
khuyến mại
giảm
giảm-giá
giảm_sâu
siêu
siêu*khuyến*mại

Hình 2


Bước 1: tiền xử lý tập từ khóa
Bước 2: tiền xử lý văn bản đầu vào
Bước 3: tìm kiếm từ khóa trong văn bản đầu vào

Hình 3