SO SÁNH VĂN BẢN DỰA TRÊN MÔ HÌNH VÉC-TƠ

COMPARISON OF THE DOCUMENTS BASED ON VECTOR MODEL

Võ Trung Hùng[1], Nguyễn Thị Ngọc Anh[1], Hồ Phan Hiếu[1], Nguyễn Ngọc Huyền Trân[2], Võ Duy Thanh[2]

[1]*Đại học Đà Nẵng; vthung@dut.udn.vn, ntnanh@ued.udn.vn, hophanhieu@ac.udn.vn*

[2]*Trường Cao đẳng CNTT Hữu nghị Việt - Hàn; nguyenngochuyentran84@gmail.com, thanhvd59@gmail.com*

Tóm tắt - Trong bài báo này, chúng tôi trình bày các kết quả nghiên cứu liên quan đến việc so sánh mức độ giống nhau của hai văn bản. Việc so sánh này phục vụ mục đích xác định mức độ giống nhau của một văn bản này với một văn bản khác. Phương pháp của chúng tôi đề xuất là chuyển các văn bản thành các véc-tơ. Mỗi phần tử của véc-tơ là trọng số tương ứng với từ chỉ mục xuất hiện trong văn bản. Việc so sánh mức độ giống nhau của hai văn bản được chuyển về tính góc tạo bởi hai véc-tơ. Góc này đặc trưng cho mức độ giống/khác nhau giữa hai văn bản. Chúng tôi đã phát triển công cụ phục vụ so sánh hai văn bản hoặc một văn bản với một tập n văn bản cho trước. Kết quả đạt được phản ánh đúng mức độ giống/khác nhau và đáp ứng mục tiêu đặt ra.

Từ khóa - mô hình véc-tơ; so sánh văn bản; phát hiện sao chép; độ đo; véc-tơ hóa

Abstract - In this paper, we present the result of the study related to the comparability of two documents. This comparison aims to determine the similarity of a text/document with an other one. Our method is converting a document into a vector. Each element of vector is a weight corresponding to the index term that appears in the text. The similarity comparison of the two texts are transformed into angles created by two vectors. This angle represents the similarity/difference between the two documents. We have developed a tool that compares a document with two or a set of documents. The results reflect exactly the similarity/difference and the achievement of the objectives.

Key words - vector model; document comparison; copy detection; measurement; vectorization

1. Giới thiệu

Cùng với sự phát triển của Internet, hoạt động trao đổi, chia sẻ tài liệu cũng diễn ra phổ biến. Các bài báo, tài liệu nghiên cứu, báo cáo thực tập, khóa luận tốt nghiệp, luận văn,… được phổ biến trên mạng Internet ngày càng nhiều. Người sử dụng có thể tìm thấy những thông tin cần thiết tương đối nhanh và dễ dàng. Tuy nhiên, bên cạnh ưu điểm là cung cấp một nguồn tài liệu tham khảo phong phú thì tình trạng đạo văn đang trở thành một vấn nạn. Bài toán đặt ra là làm thế nào để phát hiện việc sao chép văn bản, để chất lượng các bài báo cáo, khóa luận, luận văn ngày càng cao.

Hiện nay, những nghiên cứu phát hiện sự trùng lặp trên các văn bản đã cho ra đời nhiều công cụ hiệu quả và có thể sử dụng trực tuyến như Plagiarism Checker Software, Turnitin,... Nhưng những hệ thống này chỉ cho phép phát hiện sự trùng lặp của dữ liệu có trong tên miền gốc và chỉ thực hiện trực tuyến trên môi trường Internet và dành cho các tài liệu tiếng Anh. Bên cạnh đó, việc mở rộng cơ sở dữ liệu mẫu theo yêu cầu người sử dụng trở nên khó khăn và tốn chi phí rất cao. Vì thế, cần tiếp tục nghiên cứu để tìm kiếm các giải pháp tốt hơn. Hiện tại, có rất nhiều thuật toán so khớp hai văn bản được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau như: tìm kiếm thông tin, phát hiện đột nhập trong an ninh mạng, tìm mẫu trong chuỗi ADN,… Mỗi thuật toán so khớp có một hướng tiếp cận khác nhau và mỗi thuật toán đều có những ưu điểm và hạn chế riêng. [1]

Trong bài báo này, chúng tôi tập trung nghiên cứu, cải tiến giải thuật so sánh văn bản dựa trên mô hình véc-tơ. Để phát hiện trên văn bản D_1 có sao chép từ văn bản D_2 hay không thì cách làm là chuyển D_1 thành véc-tơ n chiều mà mỗi chiều của véc-tơ có thể là một từ, một câu hoặc một đoạn trong văn bản D_1. Tương tự, chuyển văn bản D_2 thành véc-tơ m chiều và sau đó so sánh 2 véc-tơ với nhau. Mô hình véc-tơ này phù hợp với bài toán phát hiện sao chép. Chúng ta có thể mở rộng để đánh giá mức độ giống nhau của một văn bản với nhiều văn bản khác đã có.

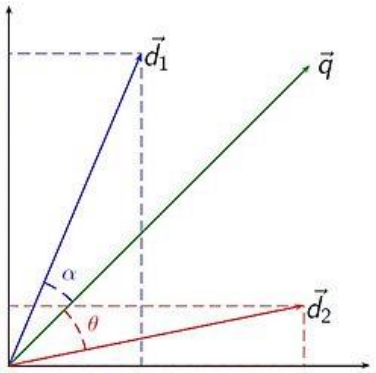
Nội dung bài báo được tổ chức thành 5 phần. Phần thứ nhất trình bày lý do nghiên cứu và giới thiệu về phương pháp, kết quả đạt được. Phần thứ 2 trình bày một số kết quả nghiên cứu đã có liên quan đến bài báo gồm mô hình véc-tơ và so khớp văn bản. Phần thứ 3 giới thiệu nội dung giải pháp do chúng tôi đề xuất liên quan đến mô hình tổng quát, quá trình véc-tơ hóa văn bản và một số giải thuật liên quan. Phần thứ 4 trình bày kết quả thử nghiệm và một số nhận xét trên kết quả đạt được. Phần cuối là kết luận và hướng phát triển trong tương lai.

2. Một số nghiên cứu liên quan

2.1. Mô hình véc-tơ

Mô hình véc-tơ là một mô hình đại số thông dụng và đơn giản dùng để biểu diễn văn bản. Một văn bản được mô tả bởi một tập các từ khóa hay còn gọi là các từ chỉ mục (index terms) sau khi đã loại bỏ các từ ít có ý nghĩa (stop word). Tập các từ chỉ mục xác định một không gian mà mỗi từ chỉ mục tượng trưng cho một chiều trong không gian đó. Các từ chỉ mục này cũng chính là các từ chứa nội dung chính của tập văn bản, mỗi từ chỉ mục này được gán một trọng số. Ta có thể sử dụng các phép toán trên mô hình véc-tơ để tính toán độ đo tương tự giữa văn bản truy vấn và các văn bản mẫu. [7], [9]

Ví dụ, văn bản d được biểu diễn theo dạng $\vec{d}$ với $\vec{d} \in R^m$ là một véc-tơ m chiều. Trong đó $\vec{d} = \{w_1, w_2, \dots, w_m\}$ và m là số chiều của véc-tơ văn bản d, mỗi chiều tương ứng với một từ trong tập hợp các từ, w_i là trọng số của đặc trưng thứ i (với $1 \leq i \leq m$). Sự tương tự của hai văn bản thường được định nghĩa là khoảng cách các điểm hoặc là góc giữa những véc-tơ trong không gian.



Hình 1. *Ví dụ về góc tạo bởi hai véc-tơ $\vec{d}_1$, $\vec{d}_2$ với $\vec{q}$*

2.2. So khớp chuỗi

Bài toán so khớp chuỗi được phát biểu như sau: Cho trước một chuỗi văn bản có độ dài *n* và một mẫu có độ dài *m*, hãy tìm sự xuất hiện của mẫu trong văn bản. Để tìm tất cả các sự xuất hiện của mẫu trong văn bản, thực hiện bằng cách quét qua toàn bộ văn bản một cách tuần tự. Bài toán "so khớp" có đặc trưng như một bài toán tìm kiếm, trong đó mẫu được xem như khóa.

Hiện nay, có một số thuật toán nhằm giải quyết bài toán so khớp như:

2.2.1. Thuật toán Brute-Force

Thuật toán Brute-Force là một thuật toán theo kiểu vét cạn. Bằng cách dịch chuyển biến đếm j từ trái qua phải lần lượt từng ký tự của tập tin văn bản. Sau đó lấy m ký tự liên tiếp trong s (bắt đầu từ vị trí j) tạo thành một chuỗi phụ r. So sánh r với p, nếu giống nhau thì xuất kết quả. Thực hiện lại quá trình trên cho đến khi j>n-m+1. [4]

2.2.2. Thuật toán Knuth-Morris-Pratt

Thuật toán so khớp chuỗi Knuth–Morris–Pratt (hay thuật toán KMP) tìm kiếm sự xuất hiện của một "từ" trong một "chuỗi văn bản" bằng cách tiếp tục quá trình tìm kiếm khi không phù hợp, bỏ qua quá trình kiểm tra lại các ký tự đã so sánh trước đó. Ý tưởng, ở mỗi thời điểm, thuật toán luôn được xác định bằng hai biến kiểu nguyên, n là độ dài của chuỗi s, và m là độ dài của chuỗi p. [3]

2.2.3. Thuật toán Boyer-Moore

Ý tưởng của thuật toán này là giả sử có chuỗi s và chuỗi p, cần tìm p trong s; bắt đầu kiểm tra các ký tự của p và s từ phải sang trái và khi phát hiện sự khác nhau đầu tiên, thuật toán sẽ tiến hành dịch p qua phải để thực hiện so sánh tiếp. [2]

2.2.4. Thuật toán Rabin-Karp

Thuật toán Rabin-Karp [5] sử dụng tính tương đương của hai số đồng dư với một số thứ ba (cho số nguyên dương n, hai số nguyên a, b được gọi là đồng dư theo mô-đun n nếu chúng có cùng số dư khi chia cho n). Ta có thể xem mỗi ký tự thuộc bảng chữ cái A là một số trong hệ đếm cơ số d, với d=|A|.

Với mẫu $p[1...m]$ đã cho, gọi *p* là biểu diễn số tương ứng của nó. Tương tự như thế, với văn bản $T[1...n]$, ta ký hiệu t_s là biểu diễn số của chuỗi con $T[s+1...s+m]$ có độ dài m, với $s = 0,1,..., n-m$. Hiển nhiên, $t_s = p$ nếu và chỉ nếu $T[s+1...s+m] = p[1...m]$.

2.2.5. Nhận xét

Ta nhận thấy việc tìm kiếm bằng Brute–Force có thể là rất chậm đối với một số mẫu nào đó, ví dụ nếu chuỗi cần xét là một chuỗi nhị phân. Trong trường hợp xấu nhất là khi tất cả mẫu thử đều là số 0 và kết thúc bởi một số 1. Mà với mỗi vị trí $n-m+1$ vị trí đều có thể khớp với nhau, tất cả các ký tự trên mẫu đều được so sánh với từng ký tự văn bản, do đó cần phải thực hiện $n-m+1$ phép so sánh. Mặt khác, thường thì m rất nhỏ so với n, như vậy số phép so sánh ký tự xấp xỉ bằng $m * n$.

Thuật toán Knuth–Morris–Pratt dùng ít phép toán so sánh hơn Brute–Force. Tuy nhiên, trong ứng dụng thực tế thì thuật toán Knuth–Morris–Pratt nhanh hơn không đáng kể so với thuật toán Brute–Force. Thuật toán Knuth–Morris–Pratt thực hiện tìm kiếm tuần tự trong văn bản và không yêu cầu phải dự phòng văn bản đó. Điều này có ý nghĩa khi áp dụng trên một tập tin lớn, thuật toán này sẽ tiêu tốn bộ nhớ đệm ít hơn.

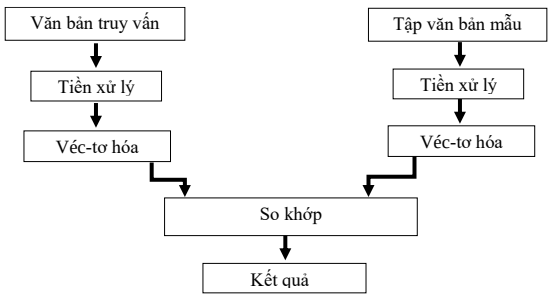
Thuật toán Boyer–Moore không dùng nhiều hơn $m+n$ phép so sánh ký tự. Trong thực tế, khi các ký tự văn bản không xuất hiện trong mẫu hoặc ngoại trừ một số ít là có mặt trong mẫu, do đó mỗi phép so sánh dẫn đến mẫu sẽ dịch sang phải m ký tự, vì vậy đối với văn bản lớn và mẫu thử không dài thì thuật toán phải dùng n/m bước.

Còn thuật toán Rabin–Krap gần như là tuyến tính. Số phép so sánh theo thuật toán này là m+n, thuật toán chỉ đi tìm một vị trí trong văn bản có cùng giá trị mảng băm với mẫu.

3. Giải pháp đề xuất

3.1. Mô hình tổng quát

Quá trình so sánh một văn bản truy vấn với tập các văn bản mẫu được thực hiện theo mô hình sau:



Hình 2. *Mô hình so sánh 2 văn bản*

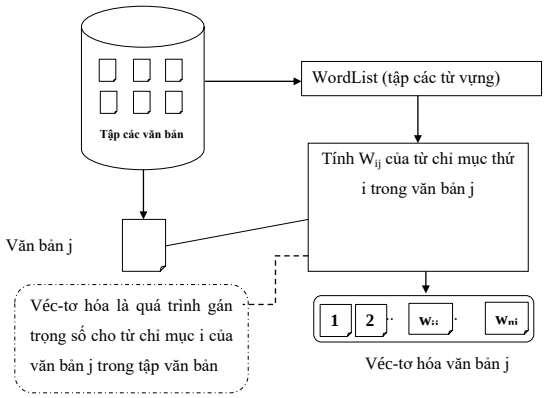
Theo mô hình này, tập các văn bản mẫu phải được xử lý [8] và véc-tơ hóa để lưu trữ. Sau đó, mỗi văn bản cần so sánh với các văn bản mẫu cũng sẽ được xử lý, véc-tơ hóa và so sánh với dữ liệu lưu trữ để phát hiện mức độ giống nhau (sao chép) từ văn bản truy vấn với tập văn bản mẫu.

3.2. Mô hình véc-tơ hóa

Trong quá trình so sánh, bước xử lý véc-tơ hóa nhằm mục đích biểu diễn các văn bản dưới dạng véc-tơ để phục vụ cho việc so sánh sau này.

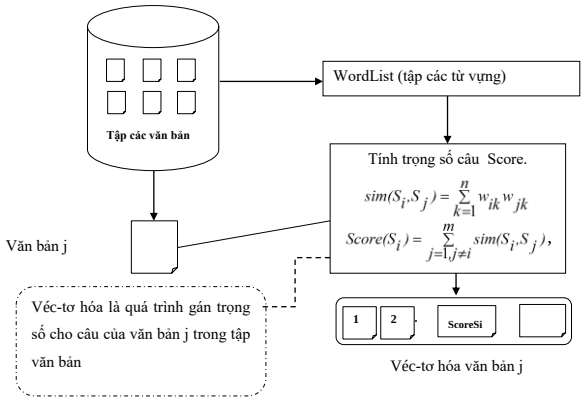
Việc véc-tơ hóa có thể thực hiện dựa trên đơn vị xử lý là từ (mỗi phần tử véc-tơ là từ) hoặc đơn vị câu (mỗi phần tử véc-tơ là một câu).

Qui trình véc-tơ hóa theo đơn vị từ được thực hiện như sau:



Hình 3. *Quá trình véc-tơ hóa theo đơn vị là từ*

Qui trình véc-tơ hóa theo đơn vị câu được thực hiện như sau:



Hình 4. *Quá trình véc-tơ hóa theo đơn vị là câu*

3.3. Các giải thuật so khớp

3.3.1. So khớp trên mô hình véc-tơ đơn vị là từ

Mục đích của thuật toán tính mức độ giống nhau của văn bản đánh giá với tập văn bản mẫu cho trước dựa trên đơn vị là từ.

Đầu vào: Tập văn bản cho trước và văn bản/đoạn văn bản cần đánh giá (đã qua quá trình tiền xử lý).

Đầu ra: tỉ lệ giống nhau giữa văn bản đánh giá với tập văn bản cho trước.

Giải thuật:

Bước 1: Tiền xử lý

- Định dạng văn bản về dạng văn bản thuần túy dạng (txt).
- Tách từ.
- Tạo danh sách từ vựng Wordlist.
- Loại bỏ StopWord.

Bước 2:

- Tính trọng số các từ chỉ mục W=TF*IDF*N với N = 1.
- Tính trọng số cục bộ L_{ij}:

$$TF = \begin{cases} 1+\log f_{ij} & if \quad f_{ij} > 0 \\ 0 & if \quad f_{ij} = 0 \end{cases}$$

- Tính f_{ij}: số lần xuất hiện của từ i trong văn bản j.

```
TF(i, j) // i: từ chỉ mục i; j: tài liệu j
  if fij= 0 then return(0)
  else return(1 + log(fij))
```

- Tính trọng số toàn cục G_i:

$$IDF = \log\left(\frac{N}{n_i}\right)$$

```
IDF(i, docs ) // docs là tập các tài liệu.
   Tính N: số văn bản trong tập văn bản.
   Tính ni: số văn bản có từ i xuất hiện.
   if (ni>0) then return log(N/ni)
   else return 0
```

Bước 3: Xây dựng ma trận trọng số để tính độ đo tương tự ngữ nghĩa giữa 2 văn bản.

Bước 4: Tính độ đo tương tự ngữ nghĩa giữa 2 văn bản.

$$\cos\theta = \frac{d_j{}^T q}{\|d_j\|_2 \|q\|_2} = \frac{\sum_{i=1}^{m} w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^{m} w_{ij}{}^2}\sqrt{\sum_{i=1}^{m} w_{iq}{}^2}}$$

Bước 5: Xây dựng ma trận trọng số tính độ đo tương đồng thứ tự giữa 2 văn bản.

Bước 6: Tính độ đo tương đồng thứ tự giữa 2 văn bản.

$$S_{r_j} = 1 - \frac{\|r_j - r_q\|}{\|r_j + r_q\|} = \frac{\sqrt{\sum_{i=0}^{m-1}\left(r_{ij} - r_{iq}\right)^2}}{\sqrt{\sum_{i=0}^{m-1}\left(r_{ij} + r_{iq}\right)^2}}$$

Bước 7: Tính độ đo giống nhau hoàn toàn giữa hai văn bản.

$$S(d_j, q) = \delta S_{\cos\theta_j} + (1-\delta) S_r = \delta\left(\frac{d_j{}^T q}{\|d_j\|_2 \|q\|_2}\right) + (1-\delta)\left(\frac{\|r_j - r_q\|}{\|r_j + r_q\|}\right)$$

3.3.2. So khớp trên mô hình véc-tơ đơn vị là câu

Mục đích của thuật toán tính mức độ giống nhau của văn bản đánh giá với tập văn bản mẫu cho trước dựa trên đơn vị là câu.

Đầu vào: Tập văn bản cho trước và văn bản/đoạn văn bản cần đánh giá (đã qua quá trình tiền xử lý).

Đầu ra: tỉ lệ giống nhau giữa văn bản đánh giá với tập văn bản cho trước.

Bước 1: Tiền xử lý

Bước 2: Tính trọng số của các từ trong câu của văn bản truy vấn:

$$w_{qk} = TF \times IDF = TF \times \log\frac{N}{n_k}$$

$$w_{jk} = TF \times IDF = TF \times \log\frac{N}{n_k}$$

Tính trọng số của câu trong văn bản truy vấn:

$$sim(S_q, S_j) = \sum_{k=0}^{n-1} w_{qk} w_{jk}$$

$$Score(S_q) = asim(S_q) = \sum_{j=0, j\neq q}^{m-1} sim(S_q, S_j)$$

Bước 3: Tính trọng số của các từ trong câu của văn bản mẫu:

$$w_{ik} = TF \times IDF = TF \times \log\frac{N}{n_k}$$

$$w_{jk} = TF \times IDF = TF \times \log\frac{N}{n_k}$$

Bước 4: Tính trọng số của câu trong văn bản mẫu:

$$sim(S_i, S_j) = \sum_{k=0}^{n-1} w_{ik} w_{jk}$$

$$Score(S_i) = asim(S_i) = \sum_{j=0, j\neq i}^{m-1} sim(S_i, S_j)$$

Bước 5: Xây dựng ma trận trọng số để tính độ đo tương tự ngữ nghĩa giữa 2 văn bản.

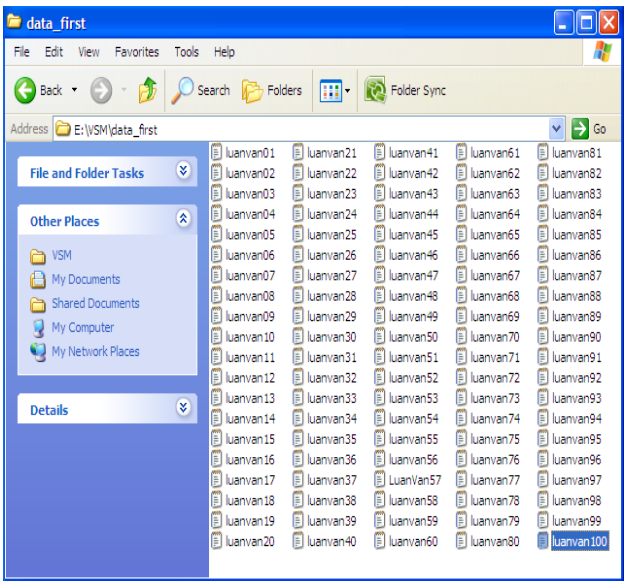
Bước 6: Tính độ đo tương tự ngữ nghĩa giữa 2 văn bản:

$$\cos\theta = \frac{d_j^T q}{\|d_j\|_2 \|q\|_2} = \frac{\sum_{i=0}^{m-1} w_{ij} w_{iq}}{\sqrt{\sum_{i=0}^{m-1} w_{ij}^2} \sqrt{\sum_{i=0}^{m-1} w_{iq}^2}}$$

4. Thử nghiệm và đánh giá

Để thử nghiệm, chúng tôi đã xây dựng một phần mềm trên C# với các chức năng cơ bản như tiền xử lý văn bản, véc-tơ hóa văn bản và so khớp. Dữ liệu phục vụ thử nghiệm là hơn 100 luận văn tốt nghiệp của sinh viên Khoa Công nghệ Thông tin, Trường Đại học Bách khoa, Đại học Đà nẵng. Những luận văn này sẽ được xử lý để giữ lại phần nội dung văn bản (text), những nội dung khác sẽ bị loại bỏ (hình ảnh, bảng số liệu,…).

Dữ liệu sau khi chuyển về dạng text:



Hình 5. *Dữ liệu là tập các văn bản mẫu*

Tỉ lệ giống nhau giữa các tài liệu được thống kê như sau:

Ký hiệu văn bản	Tỉ lệ giống nhau so theo từ	Tỉ lệ giống nhau so theo câu	Ký hiệu văn bản	Tỉ lệ giống nhau so theo từ	Tỉ lệ giống nhau so theo câu	Ký hiệu văn bản	Tỉ lệ giống nhau so theo từ	Tỉ lệ giống nhau so theo câu
vb01	100%	100%	vb35	45%	53%	vb69	38%	48%
vb02	51%	66%	vb36	44%	53%	vb70	39%	47%
vb03	33%	33%	vb37	50%	59%	vb71	41%	50%
vb04	41%	51%	vb38	48%	55%	vb72	42%	52%
vb05	42%	52%	vb39	45%	54%	vb73	34%	41%
vb06	43%	51%	vb40	49%	58%	vb74	39%	51%
vb07	44%	55%	vb41	46%	55%	vb75	38%	51%
vb08	42%	50%	vb42	34%	52%	vb76	44%	53%
vb09	45%	53%	vb43	40%	51%	vb77	39%	54%
vb10	43%	52%	vb44	28%	31%	vb78	46%	59%
vb11	45%	53%	vb45	39%	53%	vb79	42%	52%
vb12	44%	56%	vb46	43%	50%	vb80	43%	55%
vb13	44%	53%	vb47	42%	50%	vb81	35%	48%
vb14	46%	57%	vb48	45%	55%	vb82	37%	45%
vb15	47%	55%	vb49	47%	57%	vb83	40%	52%
vb16	48%	55%	vb50	47%	58%	vb84	33%	44%
vb17	44%	52%	vb51	46%	58%	vb85	24%	33%
vb18	47%	54%	vb52	46%	56%	vb86	35%	46%
vb19	43%	53%	vb53	48%	56%	vb87	29%	24%
vb20	47%	57%	vb54	46%	57%	vb88	26%	34%
vb21	44%	54%	vb55	46%	54%	vb89	13%	20%
vb22	44%	53%	vb56	38%	35%	vb90	29%	46%
vb23	46%	54%	vb57	46%	52%	vb91	26%	36%
vb24	44%	52%	vb58	42%	54%	vb92	37%	46%
vb25	44%	56%	vb59	34%	49%	vb93	32%	45%
vb26	46%	57%	vb60	41%	50%	vb94	33%	49%
vb27	47%	54%	vb61	47%	60%	vb95	36%	52%
vb28	48%	55%	vb62	44%	56%	vb96	39%	50%
vb29	46%	55%	vb63	44%	57%	vb97	28%	46%
vb30	45%	55%	vb64	48%	60%	vb98	40%	52%
vb31	42%	52%	vb65	52%	61%	vb99	44%	57%
vb32	44%	54%	vb66	39%	48%	vb100	47%	56%
vb33	45%	55%	vb67	40%	50%			
vb34	41%	49%	vb68	46%	53%			

Hình 6. *Thống kê tỉ lệ giống nhau của văn bản 1 với các văn bản khác trong kho dữ liệu*

Qua thử nghiệm, chúng tôi nhận thấy kết quả tỉ lệ so khớp có sự chênh lệch giữa véc-tơ hóa văn bản dựa trên từ và véc-tơ hóa văn bản dựa trên câu. Sự chênh lệnh này là do phụ thuộc vào phương pháp và các hàm tính trọng số.

Kết quả so sánh có giá trị là 100% khi hai văn bản giống nhau hoàn toàn và kết quả là là 0% khi hai văn bản không có bất kỳ từ vựng nào giống nhau (khác nhau hoàn toàn). Để có kết quả tỉ lệ chuẩn nhất khi các văn bản có sự chênh lệch về độ dài không quá lớn. Ví dụ, khi so khớp đoạn văn bản truy vấn với văn bản mẫu, nếu văn bản mẫu có kích thước lớn và đoạn văn bản trong văn bản mẫu giống đoạn văn bản truy vấn mà chỉ chiếm khoảng 16% trong văn bản mẫu, thì kết quả so khớp chênh lệch chạy từ 14% - 20%.

Thời gian và dung lượng tiêu tốn cho quá trình so khớp phụ thuộc vào độ dài của văn bản so khớp (số lượng từ vựng có trong văn bản).

5. Kết luận

Trong bài báo này, chúng tôi đã sử dụng một số kỹ thuật của xử lý ngôn ngữ tự nhiên, mô hình véc-tơ để biểu diễn văn bản, các thuật toán so khớp mẫu, ngôn ngữ C# và cơ sở dữ liệu bán cấu trúc dưới dạng XML để thực hiện các nghiên cứu và thử nghiệm về đánh giá sự giống nhau giữa các văn bản.

Chúng tôi đã phát triển công cụ và thử nghiệm để phát hiện sao chép trên văn bản thông qua việc sử dụng mô hình véc-tơ. Công cụ cho phép kiểm tra 2 văn bản bất kỳ, 2 đoạn văn bản bất kỳ, đoạn văn bản với văn bản, một văn bản với nhiều văn bản có sao chép với nhau hay không. Ứng dụng được thử nghiệm trên một tập 100 luận văn tốt nghiệp thuộc lĩnh vực công nghệ thông tin.

Trong thời gian đến, chúng tôi sẽ tiếp tục các nghiên cứu liên quan như: cải tiến mô hình véc-tơ để hạn chế số

lượng chiều cho văn bản khi véc-tơ hóa; tích hợp các công cụ tiền xử lý vào trong ứng dụng; nghiên cứu các giải pháp mới, đặc biệt là các kết quả nghiên cứu trong lĩnh vực sinh học, vào bài toán phát hiện sao chép.

TÀI LIỆU THAM KHẢO

[1] J.-I. Aoe, "Computer algorithms: string pattern matching strategies", *IEEE Computer Society Press*, 1994, pp. 97-107.

[2] A. Postolico, R. Giancarlo, "The Boyer-Moore-Galil string searching strategies revisited", *SIAM Journal on Computing*, 1986, pp. 98-105.

[3] M. Crochemore, C. Hancart, T. Lecroq, *"*Algorithms on Strings*"*, *Cambridge University Press*, 1997, pp. 1-58.

[4] M. Crochemore, C. Hancart, *"*Pattern Matching in Strings*"*, in *Algorithms and Theory of Computation Handbook*, 1999, pp. 11-28.

[5] D. Knuth, J.H. Morris, V. Pratt, *"*Fast pattern matching in strings*"*, *SIAM Journal on Computing*, 1977, p.p 323–350.

[6] E. Chisholm and T.G. Kolda, *"*New Term Weighting Formulas For The Vector Space Method In Information Retrieval*"*, Oak Ridge, 1999, pp. 31-63.

[7] G. Salton, A. Wong, C. S. Yang, *"*A vector space model for automatic indexing*"*, Commun. ACM, 18, 1975, pp. 613-620.

[8] L. H. Phuong and H. T. Vinh, *"*A Maximum Entropy Approach to Sentence Boundary Detection of Vietnamese Texts*"*, *IEEE International Conference on Research, Innovation and Vision for the Future RIVF 2008*, 2008, pp. 102-122.

[9] N. Polettini, *"*The Vector Space Model in Information Retrieval-Term Weighting Problem*"*, Sommarive 14, 2004, pp. 69-91.

(BBT nhận bài: 16/03/2017, hoàn tất thủ tục phản biện: 26/03/2017)