MỘT SỐ PHƯƠNG PHÁP TÍNH ĐỘ TƯƠNG ĐỒNG VĂN BẢN DỰA TRÊN MÔ HÌNH VEC-TƠ

SIMILARITY MEASUREMENTS OF TEXTUAL DOCUMENTS BASED ON VECTOR MODEL

Hồ Phan Hiếu, Võ Trung Hùng, Nguyễn Thị Ngọc Anh

Đại học Đà Nẵng; hophanhieu@ac.udn.vn, vthung@dut.udn.vn, ngocanhnt@ued.udn.vn

Tóm tắt - Trong bài báo, nhóm tác giả trình bày các kết quả nghiên cứu liên quan đến việc biểu diễn văn bản theo mô hình vec-tơ, sau đó ứng dụng các độ đo để tính khoảng cách giữa hai vec-tơ để biết được độ tương đồng của hai văn bản và độ tương đồng của văn bản truy vấn so với tập văn bản mẫu. Phương pháp của nhóm tác giả đề xuất là chuyển các văn bản thành các vec-tơ. Mỗi phần tử của vec-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 khoảng cách giữa hai vec-tơ qua các độ đo Cosine, Jaccard, Matthanan, Levenshtein. Kết quả cho biết được mức độ giống giữa hai văn bản. Nhóm tác giả 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 nhau của văn bản so với giá trị ước lượng của tập văn bản mẫu.

Từ khóa - độ tương đồng; mô hình vec-tơ; so khớp văn bản; đo khoảng cách vec-tơ; phát hiện sao chép

Abstract - In this paper, we first present the research results related to the representation of text in vector model, then apply some measurements to calculate the distance between two vectors to define the similarity of the two test textual documents and the similarity of the testing text documents versus the sample text dataset. Our proposed method is to convert text-based documents into vectors. Each element of the vector is the weight corresponding to the index text. Comparison of the two texts is shifted to the calculation of the distance between two vectors via the Cosine, Jaccard, Matthanan, Levenshtein measures. Consequently, those results denote the similarity between the two texts. We have developed a tool for comparing two texts or a abitrary document with a given document. The achieved results accurately reflect the similarity of the text versus the estimated value of the sample text set.

Key words - similarity measurement; vector model; document comparison; distance formula vectors; copy detection

1. Giới thiệu

Ngày nay, các tài liệu văn bản được công khai trên mạng Internet ngày càng nhiều. Người sử dụng có thể tìm thấy những tài liệu cần thiết một cách nhanh chóng và dễ dàng. Tuy nhiên, bên cạnh ưu điểm là cung cấp một nguồn tài liệu tham khảo phong phú thì tình trạng sao chép cũng đang trở thành một vấn nạn. Giải quyết bài toán phát hiện sao chép cần có những nghiên cứu liên quan đến tính toán độ tương đồng để có thể đánh giá được mức độ giống nhau của văn bản.

Hiện nay, đã có nhiều công trình nghiên cứu về tính toán độ tương đồng văn bản, được ứng dụng hữu ích trong rất nhiều lĩnh vực như tìm kiếm, dịch tự động, trích chọn thông tin, tóm tắt văn bản, khai phá văn bản, web ngữ nghĩa, học máy, phát hiện sao chép văn bản, … [1-2]. Giải quyết bài toán cơ bản về tính toán độ tương đồng văn bản thường dựa trên mô hình vec-tơ. Trong bài báo này, nhóm tác giả tập trung nghiên cứu cách biểu diễn văn bản theo mô hình vec-tơ, sau đó ứng dụng các độ đo như: Cosine, Jaccard, Matthanan, Levenshtein để tính khoảng cách giữa hai vec-tơ để biết được độ tương đồng của hai văn bản. Bằng cách tương tự, nhóm tác giả cũng tính được độ tương đồng giữa một văn bản cần kiểm tra với tập văn bản đã có trong kho dữ liệu. Qua kết quả nghiên cứu và thực nghiệm, có thể thấy được mô hình vec-tơ là một phương pháp phù hợp để biểu diễn văn bản trong tính toán độ tương đồng văn bản.

2. Một số nghiên cứu liên quan

2.1. Phương pháp biểu diễn văn bản

Trong xử lý văn bản có rất nhiều phương pháp có cách tính toán khác nhau, nhưng nhìn một cách tổng quan thì các phương pháp đó thường không tương tác trực tiếp trên tập dữ liệu thô ban đầu, mà thường thực hiện một số bước như tiền xử lý văn bản và mô hình hóa văn bản.

Văn bản trước khi được mô hình hóa, tức là trước khi sử dụng, cần phải được tiền xử lý. Quá trình tiền xử lý sẽ giúp nâng cao hiệu suất và giảm độ phức tạp khi sử dụng các phương pháp tính độ tương đồng. Tùy vào mục đích khai thác mà chúng ta sẽ có những phương pháp tiền xử lý văn bản khác nhau như: chuyển hẳn về chữ thường; loại bỏ các ký tự đặc biệt, các chữ số, phép toán số học; tách văn bản thành các câu và các từ riêng lẻ để sử dụng cho mục đích tính toán sau này; loại bỏ các từ dừng (stopword); lưu các câu và các từ vào một cấu trúc dữ liệu phù hợp;… Nói cách khác, một văn bản ở dạng thô (dạng chuỗi) cần được chuyển sang một mô hình khác để tạo thuận lợi cho việc biểu diễn và tính toán. Tùy thuộc vào từng thuật toán xử lý khác nhau mà chọn mô hình biểu diễn phù hợp.

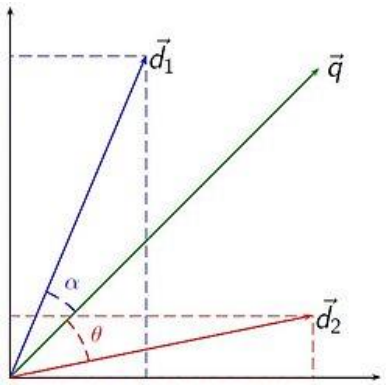
Trong các cơ sở dữ liệu văn bản, mô hình vec-tơ là mô hình biểu diễn văn bản được sử dụng phổ biến nhất. Mối quan hệ giữa các tập văn bản được thực hiện thông qua việc tính toán trên các vec-tơ biểu diễn nên đem lại hiệu quả cao. Theo mô hình này, mỗi văn bản được biểu diễn thành một vec-tơ. Mỗi thành phần của vec-tơ là một từ khóa riêng biệt trong tập văn bản gốc và được gán một giá trị là hàm f, chỉ mật độ xuất hiện của từ (hay từ khóa) trong văn bản [3]. Trong nghiên cứu này, nhóm tác giả đề xuất cách biểu diễn văn bản theo mô hình vec-tơ với ma trận trọng số từ (hay từ khóa)/tài liệu theo từ hoặc n-gram từ và sử dụng một số độ đo để tính toán, đo khoảng cách giữa các vec-tơ để đưa ra độ tương đồng giữa các văn bản.

2.2. Mô hình vec-tơ

Mô hình vec-tơ là một mô hình đại số thông dụng và đơn giản, thường được dùng để biểu diễn văn bản. Sau khi tiền xử lý, một văn bản được mô tả bởi một tập các từ hay cụm từ (gọi là từ chỉ mục). 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 [4]. Để tính toán độ đo tương đồng giữa văn bản truy vấn và các văn bản mẫu, chúng ta có thể sử dụng các phép toán trên mô hình vec-tơ.

Với một văn bản d được biểu diễn dưới dạng $\vec{d}$ với $\vec{d} \in R^m$ là một vec-tơ m chiều. Trong đó $\vec{d} = \{w_1, w_2, \dots, w_m\}$ và m là số chiều của vec-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 $1 \leq i \leq m$). Độ tương tự của hai văn bản thường được định nghĩa là khoảng cách các điểm hoặc là góc giữa những vec-tơ trong không gian, được minh họa như Hình 1.



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

2.3. So khớp chuỗi

Với bài toán so khớp văn bản, có thể giải quyết thông qua việc 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 sự xuất hiện của tất cả các 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. Một số thuật toán để giải quyết bài toán so khớp chuỗi như: Brute-Force, Morris-Pratt, Knuth-Morris-Pratt (KMP), Boyer-Moore, Karp-Rabin, Horspool, … [5-6]. Những thuật toán này tập trung vào vấn đề so sánh hai chuỗi ký tự bất kỳ và phát hiện sự giống nhau giữa chúng.

Trong một số trường hợp, việc đo độ tương đồng giữa hai đoạn văn bản là sử dụng so khớp từ đơn giản và tạo ra một điểm tương tự trên số đơn vị từ vựng xảy ra ở cả hai đoạn văn bản đầu vào. Việc loại bỏ các từ dừng, gán nhãn từ loại, so khớp tập con dài nhất, cũng như gán các trọng số, … đều có thể được tích hợp để mang lại hiệu quả cho phương pháp so khớp văn bản.

Qua quá trình khảo sát, nhóm tác giả nghiên cứu các thuật toán so khớp chuỗi để có thể ứng dụng trong bài toán tính độ tương đồng văn bản.

2.4. Các độ đo tương đồng

Sự tương đồng giữa hai văn bản là sự giống nhau về nội dung giữa hai văn bản đó. Do đó, hai văn bản là bản sao hoặc gần giống nhau thì sẽ có nội dung giống nhau nhiều, hay "độ tương đồng" giữa hai văn bản là cao. Độ tương đồng nằm trong khoảng giữa 0 và 1, như vậy độ tương đồng càng gần 1 thì khả năng các văn bản là bản sao hoặc gần giống nhau là cao, và ngược lại. Do đó, để xét xem các văn bản có phải là bản sao hoặc gần giống nhau hay không ta phải tính độ tương đồng giữa chúng [7].

Một số phương pháp tính độ tương đồng văn bản có kết quả khả quan trong tiếng Anh như sử dụng các tập dữ liệu chuẩn về ngôn ngữ để tìm ra mối quan hệ giữa các từ trong các kho dữ liệu như: Wordnet, Brown Corpus, Penn TreeBank, … Phương pháp dựa trên tập dữ liệu và dựa trên tri thức sẽ xác định sự giống nhau về mặt ngữ nghĩa của từ, phương pháp dựa trên chuỗi xác định sự giống nhau về mặt từ vựng. Trong đó, phương pháp thao tác trên chuỗi xác định sự giống nhau giữa các chuỗi dựa vào thành phần cấu tạo nên chuỗi và được tách thành hai phương pháp là dựa trên ký tự (character-based) và dựa trên từ (term-based). Một số thuật toán dựa trên ký tự như: chuỗi con chung dài nhất (LCS), Damerau-Levenshtein, Jaro, Jaro-Winkler, Needleman-Wunsch, Smith-Waterman, n-gram và thuật toán dựa trên từ như: khoảng cách Manhattan, Cosine Similarity, hệ số Dice, khoảng cách Euclid, Jaccard Similarity, hệ số Matching và hệ số Overlap [8].

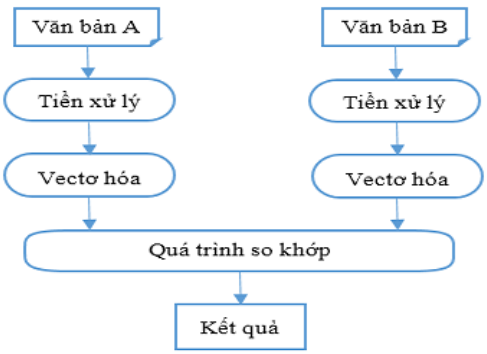
Trong nghiên cứu này, nhóm tác giả nghiên cứu, cài đặt và thực nghiệm dựa trên bốn độ đo Cosine, Jaccard, Matthanan, Levenshtein để tính toán mức độ giống nhau của văn bản tiếng Việt.

3. Giải pháp đề xuất

Trong phần này, nhóm tác giả trình bày giải pháp để thực hiện mô hình hóa văn bản thành vec-tơ và tính toán độ tương tự bằng cách đo khoảng cách giữa các vec-tơ và hiển thị kết quả là tỉ lệ giống nhau của hai văn bản hoặc của một văn bản với tập văn bản trong kho dữ liệu.

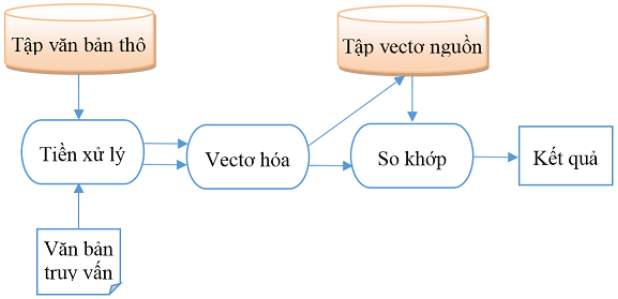
3.1. Mô hình tổng quát

Quá trình so sánh hai văn bản: Với hai văn bản đầu vào, qua quá trình tiền xử lý, tiếp đến là vectơ hóa để biểu diễn văn bản thành dạng vectơ, sau đó thực hiện so khớp giữa hai vectơ để cho kết quả là độ tương đồng giữa hai văn bản [6]. Các quá trình này được thực hiện theo mô hình đề xuất như sau:



Hình 2. *Mô hình so sánh hai văn bản*

Quá trình so sánh giữa một văn bản với tập văn bản nguồn: Nhóm tác giả đề xuất theo mô hình dưới đây, tập các văn bản nguồn phải được xử lý và vec-tơ hóa để lưu trữ. Sau đó, mỗi văn bản cần so sánh với các văn bản nguồn cũng sẽ được xử lý, vec-tơ hóa và so sánh với dữ liệu lưu trữ để phát hiện mức độ giống nhau từ văn bản truy vấn với tập văn bản nguồn.

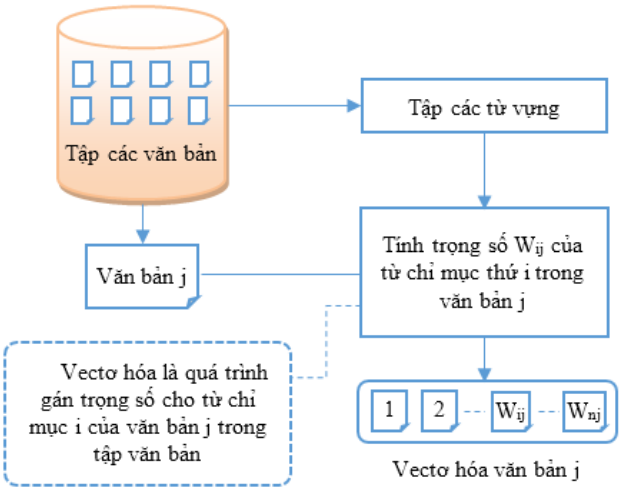


Hình 3. *Mô hình so sánh văn bản với tập văn bản nguồn*

3.2. Mô hình vec-tơ hóa

Trong quá trình so sánh, bước xử lý vec-tơ hóa nhằm mục đích biểu diễn các văn bản dưới dạng vec-tơ để phục vụ cho việc so sánh sau này. Việc vec-tơ hóa có thể thực hiện dựa trên đơn vị xử lý là từ (mỗi phần tử vec-tơ là từ) hay n-gram từ.

Quy trình vec-tơ hóa theo từ được thực hiện như sau:



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

Để tính giá trị đặc trưng cho văn bản, nhóm tác giả thực hiện bằng phương pháp TF*IDF qua các bước như sau:

+ Bước 1 - Tính f_{ij}: Số lần xuất hiện của từ chỉ mục thứ i trong văn bản j.

+ Bước 2 - Tính trọng số cục bộ:

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

+ Bước 3 - Tính trọng số toàn cục:

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

+ Bước 4 - Tính trọng số các từ chỉ mục: W_{ij}=TF*IDF.

3.3. Các phương pháp đo độ tương đồng

Trong phần này, nhóm tác giả giới thiệu, nêu ý tưởng của thuật toán và các bước thực hiện để tính độ tương đồng văn bản thông qua bốn độ đo tương đồng Cosine, Jaccard, Matthanan, Levenshtein. Các mục nội dung đầu đề cập đến việc tính độ tương đồng của hai văn bản. Phần sau, bằng cách tương tự sẽ tính được độ tương đồng giữa một văn bản truy vấn với tập văn bản nguồn đã có trong kho dữ liệu.

3.3.1. Độ đo Cosine

Tính độ tương đồng văn bản dựa trên độ đo Cosine là phương pháp tương đối đơn giản và cho kết quả với độ chính xác cao. Các văn bản được biểu diễn theo mô hình túi từ (bag-of-words). Trong mô hình này, một văn bản được thể hiện như là một túi các từ của nó, không theo ngữ pháp và thứ tự từ. Mỗi văn bản được được tách ra thành các từ hay cụm từ (n-gram từ), sau đó được bỏ vào trong túi. Mỗi từ hay cụm từ được tính tổng số lần xuất hiện và được tạo thành vec-tơ n chiều, trong đó n là số phần tử trong danh sách chung các từ hay cụm từ khác nhau của các văn bản. Sau khi chuyển hai văn bản thành vec-tơ là $\vec{a}$ và $\vec{b}$, ta có thể sử dụng độ đo Cosine để tính toán độ tương đồng của các văn bản [3]. Công thức tính độ tương đồng Cosine là:

$$\text{Sim}_C(\vec{a},\vec{b}) = \frac{\vec{a} * \vec{b}}{\|\vec{a}\|\|\vec{b}\|} \quad (1)$$

Bảng 1. *Thuật toán tính độ tương đồng Cosine*

Thuật toán 1: Tính độ tương đồng Cosine	
1:	**Đầu vào:** 2 văn bản A và B.
2:	**Xử lý:** Thực hiện qua các bước sau:
3:	- Tiền xử lý (tách từ, tạo danh sách từ vựng, …)
4:	- Xây dựng tập từ vựng chung T = {t_1, t_2….}
5:	- Mô hình hóa văn bản thành vec-tơ: Dựa vào T ta tạo vec-tơ tần số từ của A và B lần lượt là $\vec{a}$ và $\vec{b}$ (bằng cách tính TF*IDF)
6:	- Từ 2 vec-tơ tần số từ tương ứng với 2 văn bản, tính cos góc giữa 2 vec-tơ bằng cách sử dụng công thức tính độ tương đồng Cosine theo (1)
7:	**Đầu ra:** Độ tương đồng giữa 2 văn bản A và B.

3.3.2. Độ đo khoảng cách Manhattan

Khoảng cách Manhattan là một dạng khoảng cách giữa hai điểm trong không gian Euclid với hệ tọa độ Descartes. Đại lượng này được tính bằng tổng chiều dài của hình chiếu của đường thẳng nối hai điểm này trong hệ trục tọa độ Descartes.

Khi hai văn bản được biểu diễn thành hai vec-tơ đặc trưng $\vec{a}$ và $\vec{b}$ ta có công thức Manhattan [9, 10] là:

$$\text{D}(\vec{a},\vec{b}) = \sum_{i=1}^{n} |a_i - b_i| \quad (2)$$

$\text{D}(\vec{a},\vec{b})$ nằm trong khoảng giữa 0 và 1, mức độ tương đồng giữa 2 vec-tơ này được tính toán bằng công thức sau:

$$\text{Sim}_M(\vec{a},\vec{b}) = 1 - \frac{\sum_{i=1}^{n} |a_i - b_i|}{n} \quad (3)$$

Bảng 2. *Thuật toán tính độ tương đồng Manhattan*

Thuật toán 2: Tính độ tương đồng Manhattan	
1:	**Đầu vào:** 2 văn bản A và B.
2:	**Xử lý:** Thực hiện qua các bước sau:
3:	- Tiền xử lý (tách từ, tạo danh sách từ vựng, …)
4:	- Xây dựng tập từ vựng chung T = {t_1, t_2….}
5:	- Mô hình hóa văn bản thành vec-tơ: Dựa vào T ta tạo vec-tơ tần số từ của A và B lần lượt là $\vec{a}$ và $\vec{b}$ (bằng cách tính TF*IDF)

6:	- Áp dụng công thức tính hệ số Manhattan theo (3)
7:	**Đầu ra:** Độ tương đồng giữa 2 văn bản A và B.

3.3.3. Độ đo sử dụng hệ số Jaccard

Các chuỗi đầu vào của từng văn bản được chuyển thành tập hợp n-gram. Cho hai tập hợp n-gram tương ứng với hai văn bản cần so sánh là A và B. Sau khi chuyển hai văn bản thành vec-tơ lần lượt là $\vec{a}$ và $\vec{b}$, hệ số Jaccard được tính như công thức sau [3], [11]:

$$\mathrm{Sim}_J(\vec{a},\vec{b}) = \frac{\left|\vec{a}\cap\vec{b}\right|}{\left|\vec{a}\cup\vec{b}\right|} \quad (4)$$

Bảng 3. *Thuật toán tính độ tương đồng Jaccard*

Thuật toán 3: Tính độ tương đồng Jaccard	
1:	**Đầu vào:** 2 văn bản A và B.
2:	**Xử lý:** Thực hiện qua các bước sau:
3:	- Tiền xử lý (tách từ, tạo danh sách từ vựng, …)
4:	- Mô hình hóa văn bản thành vec-tơ: Sử dụng n-gram để tạo vec-tơ tần số từ của A và B lần lượt là $\vec{a}$ và $\vec{b}$ (bằng cách tính TF*IDF)
5:	- Áp dụng công thức tính hệ số Jaccard theo (4)
6:	**Đầu ra:** Độ tương đồng giữa 2 văn bản A và B.

3.3.4. Độ đo khoảng cách Levenshtein

Khoảng cách Levenshtein thể hiện khoảng cách khác biệt giữa hai chuỗi ký tự. Khoảng cách Levenshtein giữa chuỗi A và chuỗi B là số bước nhỏ nhất để biến đổi chuỗi A thành chuỗi B thông qua ba phép biến đổi là: xóa một ký tự, thêm một ký tự và thay ký tự này thành ký tự khác.

Để tính toán khoảng cách Levenshtein, sử dụng thuật toán quy hoạch động, tính toán trên mảng 2 chiều (n+1)*(m+1), với n và m lần lượt là độ dài chuỗi A và B. Thuật toán chi tiết như sau [12]:

Bảng 4. *Thuật toán tính khoảng cách Levenshtein*

LevenshteinDistance (A[1, 2,…, *n*], B[1, 2,…, *m*]):	
1:	**Khởi tạo:** *d*[0…*m*, 0…*n*] // *m+1 hàng, n+1 cột*
2:	**For** *i*:= 0→ *m*
3:	*d*[i, 0]:= i
4:	**For** *j*:= 0→ *n*
5:	*d*[0, j]:= j
6:	**For** *i*:= 1→ *m*
7:	**For** *j*:= 1→ *n* {
8:	**If** A[i] = B[j] **then** cost:= 0
9:	**else** cost:= 1
10:	*d*[i, j]:= **min**(
11:	*d*[i−1, j] + 1,
12:	// *trường hợp xóa*
13:	*d*[i, j−1] + 1, // *trường hợp thêm*
14:	
15:	*d*[i−1, j−1] + cost // *trường hợp thay thế*
16:	)
	}
	Return *d*[n, m]

Khoảng cách Levenshtein là khoảng cách giữa hai chuỗi ký tự có giá trị của *d*[m, n], với s là độ dài của chuỗi dài nhất. Độ đo tương tự được tính theo công thức sau:

$$Sim_L(A,B) = 1 - \frac{d[\mathrm{m},\mathrm{n}]}{s} \quad (5)$$

Bảng 5. *Thuật toán tính độ tương đồng Levenshtein*

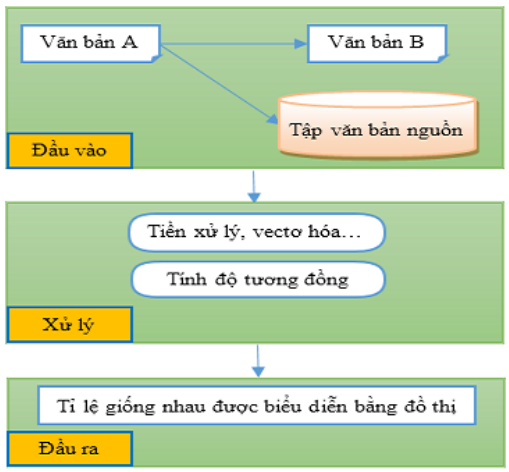
Thuật toán 4: Tính độ tương đồng Levenshtein	
1:	**Đầu vào:** 2 văn bản A và B.
2:	**Xử lý:** Thực hiện qua các bước sau:
3:	- Tiền xử lý (tách từ, tạo danh sách từ vựng, …)
4:	- Xây dựng tập từ vựng chung T = {t_1, t_2….}
5:	- Mô hình hóa văn bản thành vec-tơ: Dựa vào T ta tạo vec-tơ tần số từ của A và B lần lượt là $\vec{a}$ và $\vec{b}$ (bằng cách tính TF*IDF)
6:	- Tính khoảng cách Levenshtein theo thuật toán trên
7:	- Xác định độ dài của chuỗi dài nhất là s
8:	- Áp dụng công thức tính độ tương tự theo (5)
9:	**Đầu ra:** Độ tương đồng giữa 2 văn bản A và B.

Kiểm tra một tài liệu so với kho dữ liệu: Tương tự như quy trình kiểm tra độ tương đồng giữa hai tài liệu, đối với việc kiểm tra một tài liệu so với kho dữ liệu nhóm tác giả cũng thực hiện các bước tương tự. Về tài liệu trong kho dữ liệu, nhóm tác giả cũng thực hiện tiền xử lý, mô hình hóa các văn bản thành vec-tơ, đánh chỉ mục và lưu vào cơ sở dữ liệu để phục vụ tìm kiếm, so khớp và trích xuất thông tin của tài liệu.

Trong phần thực nghiệm, nhóm tác giả đã kiểm tra một tài liệu đầu vào so với các tài liệu trong kho dữ liệu và kết quả thực nghiệm trùng khớp với việc kiểm tra độ tương đồng giữa hai tài liệu.

3.4. Phương pháp xây dựng hệ thống

Hệ thống được xây dựng với chức năng chính là: So sánh giữa hai tài liệu và mở rộng hơn là so sánh giữa một tài liệu (văn bản truy vấn) và kho dữ liệu (tập văn bản nguồn). Dưới đây là mô hình kiến trúc của hệ thống:



Hình 5. *Mô hình kiến trúc của hệ thống*

4. Thử nghiệm và đánh giá

4.1. Tập dữ liệu thử nghiệm

Để thử nghiệm, nhóm tác giả đã xây dựng một ứng dụng với các chức năng như: tiền xử lý văn bản, vec-tơ hóa văn bản, so khớp, hiển thị kết quả và vẽ biểu đồ. Nhóm tác giả tạo bộ dữ liệu mẫu gồm các văn bản tiếng Việt bằng cách tạo hai tài liệu A và B với nội dung hoàn toàn khác nhau. Mỗi

tài liệu có 1.000 từ riêng biệt (không kể các từ dừng). Sau đó, chọn ngẫu nhiên một vài câu trong tài liệu A có 200 từ (chiếm 20% tổng số từ của tài liệu) để thay thế cho một vài câu trong tài liệu B cũng có 200 từ được chọn ngẫu nhiên. Như vậy, chúng ta có thể ước lượng chính xác tỉ lệ giống nhau giữa chúng là 20%. Tương tự, nhóm tác giả đã tạo ra các tài liệu có tỉ lệ giống nhau là 40%, 60%, 80% và 100%.

Để kiểm tra tính chính xác của các thuật toán, khi so sánh một tài liệu bất kỳ với tài liệu đầu tiên (Doc_1.docx) thì kết quả tính toán của các thuật toán được so sánh với giá trị ước lượng. Dưới đây là bảng mô tả các trường hợp thử nghiệm.

Bảng 6. *Các tài liệu mẫu để so với giá trị ước lượng*

Trường hợp thử nghiệm	Tài liệu	Ước lượng tỷ lệ giống nhau (%)
TH1	Doc_2.docx	0
TH2	Doc_3.docx	20
TH3	Doc_4.docx	40
TH4	Doc_5.docx	60
TH5	Doc_6.docx	80
TH6	Doc_7.docx	100

4.2. Kết quả thực nghiệm

Kết quả thực nghiệm đã xử lý được dữ liệu để phục vụ cho quá trình so sánh văn bản, tỉ lệ so khớp có sự chênh lệch giữa các phương pháp và so với các trường hợp ước lượng không lớn.

Các thuật toán đều cho 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 khác nhau hoàn toàn. Trong các trường hợp còn lại, kết quả của thuật toán so với giá trị ước lượng có độ chênh lệch tương đối, cụ thể là:

- Phương pháp Cosine: Thuật toán sử dụng phương pháp tần số từ, vì dữ liệu thử nghiệm được lựa chọn theo chuẩn đặt ra với độ chính xác tuyệt đối của giá trị ước lượng nên kết quả hoàn toàn chính xác. Chính vì vậy, phương pháp so sánh độ tương đồng văn bản dựa trên độ đo Cosine đem lại hiệu quả trong phát hiện sao chép văn bản khi so sánh theo từ vựng.

- Phương pháp Jaccard và Manhattan: Mặc dù hai phương pháp này có công thức tính độ tương tự khác nhau nhưng khi sử dụng tách từ đơn (n-gram bằng 1) thì cho ra hai kết quả hoàn toàn giống nhau. Nếu tách từ sử dụng bigram hoặc trigram (hoặc n lớn hơn) sẽ cho hai kết quả khác nhau dù chênh lệch không nhiều.

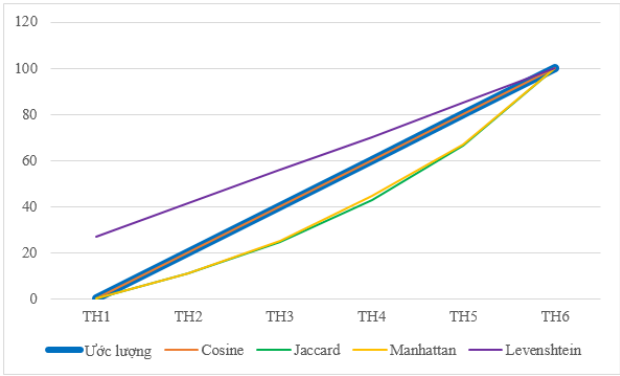
- Phương pháp Levenshtein có ưu điểm dùng để đo khoảng cách giữa hai chuỗi dựa vào ký tự, vì vậy trong trường hợp hai tài liệu khác nhau hoàn toàn về từ thì cũng có thể giống nhau về ký tự và khoảng trắng, vì thế nên độ đo tương tự của hai tài liệu khác nhau hoàn toàn cũng có thể lớn hơn 0%. Vì vậy, phương pháp này sử dụng đo độ tương tự của văn bản sẽ không hiệu quả.

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

Dưới đây là bảng tổng hợp kết quả ghi được của các phương pháp so với các trường hợp thử nghiệm và kết quả ước lượng.

Bảng 7. *Tổng hợp kết quả của các phương pháp so với các trường hợp thử nghiệm và kết quả ước lượng (với trigram)*

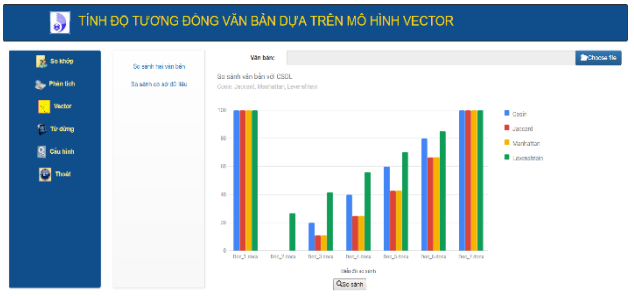
Trường hợp	Kết quả thử nghiệm tính theo độ đo (%)				Ước lượng (%)
	Cosine	Jaccard	Manhattan	Levenshtein	
TH1	0	0	0	26,86	0
TH2	20	11,01	11,11	41,72	20
TH3	40	24,91	25	56,12	40
TH4	60	42,78	42,86	70,26	60
TH5	80	66,61	66,67	85,14	80
TH6	100	100	100	100	100



Hình 6. *Biểu đồ so sánh kết quả thuật toán với tập tài liệu*

Trong thử nghiệm đối với tập tài liệu mẫu này, kết quả của phương pháp Jaccard và Manhattan gần như bằng nhau nên đường hiển thị gần bị trùng.

Hình 7. *Kết quả so sánh 2 văn bản giữa các phương pháp có độ chênh lệch khá thấp*



Hình 8. *Kết quả so sánh tỉ lệ giống nhau của một văn bản với tập văn bản mẫu*

Với kết quả trên, cho thấy nhóm tác giả đã nghiên cứu cách biểu diễn văn bản theo mô hình vec-tơ, sử dụng các độ đo để tính độ tương đồng của văn bản và xây dựng hệ thống thực nghiệm. Tuy nhiên, hệ thống so khớp mới thử nghiệm ở việc so sánh với bốn độ đo khác nhau trên các văn bản .doc hoặc .docx và kho dữ liệu nhỏ.

5. Kết luận

Trong bài báo này, nhóm tác giả đã sử dụng một số kỹ thuật của xử lý ngôn ngữ tự nhiên để xử lý văn bản, sau đó sử dụng mô hình vec-tơ để biểu diễn văn bản; nghiên cứu và ứng dụng các thuật toán so khớp mẫu, các phương pháp tính độ tương đồng để thực hiện các nghiên cứu và thử nghiệm về đánh giá sự giống nhau giữa các văn bản.

Nhóm tác giả đã phát triển công cụ và thử nghiệm để phát hiện sao chép trên văn bản thông qua việc sử dụng mô hình vec-tơ. Công cụ này cho phép kiểm tra hai văn bản bất kỳ, một văn bản với nhiều văn bản có sẵn trong kho dữ liệu có sao chép với nhau hay không. Ứng dụng được triển khai trên một bộ dữ liệu thử nghiệm và hoàn toàn có thể mở rộng, cập nhật thêm các tài liệu vào kho dữ liệu để phục vụ việc đánh giá so khớp. Thời gian xử lý của các thuật toán tương đối nhanh, với độ phức tạp tính toán không lớn, do tập dữ liệu mẫu không nhiều. Nhóm tác giả chỉ thử nghiệm và quan tâm đến kiểm tra độ chính xác của thuật toán và đưa ra một số nhận xét dựa trên kết quả thực nghiệm.

Phương pháp dựa trên vec-tơ thông thường chỉ xử lý hiệu quả trên dữ liệu nhỏ, không quá phức tạp. Hướng phát triển tiếp theo, nhóm tác giả sẽ nghiên cứu cách biểu diễn văn bản và sử dụng thuật toán so khớp phù hợp với mô hình dữ liệu để có thể giải quyết bài toán về dữ liệu lớn. Nhóm tác giả sẽ tiếp tục các nghiên cứu liên quan để cải tiến mô hình vec-tơ nhằm hạn chế số lượng chiều cho văn bản khi vec-tơ hóa, phát triển và tích hợp các công cụ hỗ trợ xử lý văn bản vào trong ứng dụng, nghiên cứu các giải pháp mới trong lĩnh vực mã hóa, xử lý chuỗi số thực, xử lý dữ liệu lớn, xử lý ngôn ngữ tiếng Việt,... để đem lại hiệu quả trong so khớp văn bản tiếng Việt.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Quỹ Phát triển KHCN Đại học Đà Nẵng trong đề tài mã số B2017-ĐN01- 07.

TÀI LIỆU THAM KHẢO

[1] Meuschke, N. and B. Gipp, "State-of-the-art in detecting academic plagiarism", *International Journal for Educational Integrity*, 9(1), 2013, pp. 50-71.

[2] Rubini, P. and M.S. Leela, "A survey on plagiarism detection in text mining", *International Journal of Research in Computer Applications and Robotics*, Vol.1, Issue 9, 2013, pp. 117-119.

[3] Huang, Anna, *Similarity measures for text document clustering*, in Proceedings of the sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand, 2008, pp. 49-56.

[4] G. Salton, A. Wong, C. S. Yang, "A vector space model for automatic indexing", *Commun. ACM*, Vol. 18, Issue 11, 1975, pp. 613-620.

[5] Singla, Nimisha, and Deepak Garg, "String matching algorithms and their applicability in various applications", *International Journal of Soft Computing and Engineering*, I(6), 2012, pp. 218-222.

[6] Hung Vo Trung, Ngoc Anh Nguyen, Hieu Ho Phan, Thi Dung Dang, "Comparison of the Documents Based On Vector Model: A Case Study of Vietnamese Documents", American Journal of Engineering Research (AJER), 2017, pp. 251-256.

[7] Reddy, G. Suresh, T. V. Rajinikanth, and A. Ananda Rao, "Clustering and Classification of Text Documents Using Improved Similarity Measure", *International Journal of Computer Science and Information Security*, Vol. 14, 2016, pp. 39-54.

[8] Gomaa, W.H. and A.A. Fahmy, "A survey of text similarity approaches", *International Journal of Computer Applications*, 68(13), 2013, pp. 13-18.

[9] Khatibsyarbini, M., et al., "A hybrid weight-based and string distances using particle swarm optimization for prioritizing test cases", *Journal of Theoretical & Applied Information Technology*, Vol. 95, Issue 12, 2017, pp. 2723-2732.

[10] Ledru, Yves, et al, "Prioritizing test cases with string distances", *Automated Software Engineering*, Vol. 19, Issue 1, 2012, pp. 65-95.

[11] Niwattanakul, Suphakit, et al, *Using of Jaccard coefficient for keywords similarity*, in Proceedings of the International MultiConference of Engineers and Computer Scientists, 2013.

[12] Su, Zhan, et al., *Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm*, Innovative Computing Information and Control, 2008 (ICICIC'08), 3rd International Conference on IEEE, 2008.

(BBT nhận bài: 10/10/2017, hoàn tất thủ tục phản biện: 22/10/2017)