Gom nhóm văn bản dựa trên mẫu hữu ích được đánh trọng phổ biến

Trần Thanh Trâm[1], Võ Đình Bảy[2*]

[1]Trường Đại học Công nghệ thông tin, Đại học Quốc gia TP Hồ Chí Minh
[2]Khoa Công nghệ thông tin, Trường Đại học Công nghệ TP Hồ Chí Minh

Ngày nhận bài 26/3/2018; ngày chuyển phản biện 30/3/2018; ngày nhận phản biện 22/4/2018; ngày chấp nhận đăng 27/4/2018

Tóm tắt:

Gom nhóm văn bản là chủ đề quan trọng trong khai thác dữ liệu văn bản, và phương pháp hiệu quả để gom nhóm văn bản là dựa trên mẫu phổ biến. Đã có nhiều thuật toán được phát triển nhằm nâng cao độ chính xác cho bài toán gom nhóm văn bản dựa trên mẫu phổ biến, nhưng lại không quan tâm tới trọng số của từ trong văn bản. Trong bài báo này, các tác giả đề xuất một phương pháp mới để gom nhóm văn bản dựa vào mẫu hữu ích được đánh trọng phổ biến thông qua việc sử dụng TF (Term Frequency) cho mỗi từ trong văn bản. Trọng số của từ trên toàn bộ tập văn bản được tính dựa vào IDF (Inverse Document Frequency), sau đó sử dụng thuật toán MWIT-FWUI để khai thác các mẫu hữu ích phổ biến. Tiếp theo, tiến hành gom nhóm văn bản bằng thuật toán MC (Maximum Capturing). Kết quả thử nghiệm trên kho ngữ liệu gồm 1.600 văn bản (16 chủ đề) cho thấy, phương pháp mới đã cải thiện đáng kể độ chính xác của việc gom nhóm văn bản so với phương pháp dựa vào mẫu phổ biến.

Từ khóa: **cơ sở dữ liệu số lượng, gom nhóm văn bản, mẫu hữu ích được đánh trọng phổ biến, mẫu phổ biến, trọng số của từ.**

Chỉ số phân loại: **1.2**

Đặt vấn đề

Hiện nay, các công trình nghiên cứu về bài toán gom nhóm văn bản tiếng Việt còn rất ít, hầu hết sử dụng các phương pháp truyền thống như phương pháp gom nhóm đồ thị [1], gom nhóm dựa trên thực thể có tên [2]…

Năm 2002, Beil và cộng sự đã giới thiệu một hướng tiếp cận mới cho bài toán gom nhóm văn bản, đó là gom nhóm văn bản dựa vào mẫu phổ biến [3]. Kết quả thực nghiệm cho thấy, gom nhóm văn bản dựa vào mẫu phổ biến có độ chính xác cao hơn, thời gian xử lý tốt hơn so với gom nhóm văn bản dựa vào bisecting k-means [4]. Công trình này mở ra một hướng tiếp cận hoàn toàn mới cho bài toán gom nhóm văn bản và là tiền đề cho rất nhiều công trình nghiên cứu sau này.

Một số thuật toán gần đây đã giải quyết được vấn đề gom nhóm văn bản dựa trên mẫu phổ biến như phương pháp CMS [5], phương pháp FIHC [6], phương pháp CFWS [7]. Tuy nhiên, các phương pháp này lại không quan tâm tới trọng số của từ trong các văn bản, chỉ quan tâm tới tần số xuất hiện của các item trong mẫu phổ biến. Trong [8], tác giả đã chỉ ra một số hạn chế của các thuật toán trên như sau: 1) Có thể dẫn tới việc trùng lặp các văn bản trong kết quả gom nhóm cuối cùng; 2) Có khuynh hướng chọn nhóm ứng viên có số văn bản ít, trong khi độ phổ biến của những văn bản đó lớn để gom thành 1 nhóm, dẫn tới kết quả một số lớn các nhóm được gom chỉ có 1 văn bản; 3) Không thể giải quyết được mâu thuẫn nhóm khi chỉ định các văn bản vào các nhóm, một văn bản có thể được chia thành nhiều nhóm khác nhau…

Khai thác mẫu phổ biến không quan tâm đến trọng số của các mục, vì vậy khi áp dụng các thuật toán khai thác mẫu phổ biến để gom nhóm văn bản, các tác giả chỉ quan tâm đến sự xuất hiện hay không xuất hiện của mỗi từ trong văn bản. Thực tế, các từ trong văn bản có số lần xuất hiện khác nhau, nên có đóng góp khác nhau. Trong bài báo này, chúng tôi đề xuất phương pháp gom nhóm văn bản dựa trên trọng số (chẳng hạn TF-IDF, tần số xuất hiện…) của các từ và thuật toán MC với mục tiêu cải thiện độ chính xác của kết quả gom nhóm.

Phương pháp nghiên cứu

Cơ sở dữ liệu (CSDL) số lượng: cho một CSDL D với một tập các giao dịch $T = \{t_1, t_2, ... t_m\}$, một tập các mặt hàng $I = \{i_1, i_2, ... i_n\}$ và một tập các trọng số $W = \{w_1, w_2, ..., w_n\}$ tương ứng với mỗi mặt hàng trong I. Với mỗi giao dịch

[*]Tác giả liên hệ: Email: bayvodinh@gmail.com

Text clustering using frequent weighted utility itemsets

Thanh Tram Tran[1], Dinh Bay Vo[2*]

[1]*University of Information Technology, Vietnam National University Ho Chi Minh City*

[2]*Faculty of Information Technology, Ho Chi Minh City University of Technology*

Received 26 March 2018; accepted 27 April 2018

Abstract:

Text clustering is one of the important topics of the text mining. One of the most effective methods for text clustering is the frequent itemsets based approach. There are many frequent itemsets based algorithms to improve the accuracy of text clustering. However, these algorithms do not focus on the weight of terms in documents. In fact, the frequency of each term in each document has a great impact on results. In this paper, we propose a new method for text clustering based on frequent weighted utility itemsets. First, we calculate TF (*Term Frequency*) for each term in documents to create a number matrix for the documents. The weights of terms in the documents is based on the IDF (*Inverse Document Frequency*). Next, we use MWIT-FWUI algorithm for mining frequent weighted utility itemsets from a number matrix and the weight of terms in the documents. Finally, based on frequent utility itemsets, we cluster text using MC (Maximum Capturing) algorithm. We have tested this method on a corpus consisting of 1,600 documents including 16 different topics. Experimental results have shown that our method significantly improves the accuracy of the text clustering compared with the clustering method using frequent itemsets.

Keywords: **frequent itemsets, frequent weighted utility itemsets, quantitative databases, text clustering, weight of terms.**

Classification number: **1.2**

$t_k = \{x_{k1}, x_{k2}, \ldots x_{kn}\}$ mà trong đó x_{ki} là số lượng sản phẩm được mua bởi mặt hàng thứ i trong giao dịch t_k.

Trọng số của từ (TF-IDF): TF-IDF (term frequency - inverse document frequency) của một từ thể hiện mức độ quan trọng của từ này trong một văn bản nằm trong một tập hợp các văn bản. Nó dựa vào tần số xuất hiện của 1 từ trong 1 văn bản (TF) và tần số nghịch của 1 từ trong tập văn bản (IDF).

TF (term frequency) là tần số xuất hiện của 1 từ trong 1 văn bản. Giá trị TF của một từ t trong văn bản d được tính theo công thức (1).

$$tf\ (t,d) = \frac{n(t,d)}{n(d)} \quad (1)$$

Trong đó, n(t, d) là số lần xuất hiện từ t trong văn bản d; n(d) là số lần xuất hiện của tất cả các từ trong văn bản d.

IDF (inverse document frequency) là tần số nghịch của một từ trong tập văn bản (corpus). Trong tập văn bản, mỗi từ chỉ có 1 giá trị IDF duy nhất được tính theo công thức (2):

$$idf\ (t,D) = log\frac{|D|}{|\{d \in D|t \in d\}|} \quad (2)$$

Trong đó, |D| là tổng số văn bản trong tập D, $|\{d \in D|t \in d\}|$ là số văn bản *d* có xuất hiện từ *t* trong tập D. Nếu từ đó không xuất hiện ở bất cứ một văn bản nào trong tập D thì mẫu số sẽ bằng 0, dẫn tới phép chia không hợp lệ, vì thế người ta thường thay bằng công thức $1+|\{d \in D|t \in d\}|$. Cơ số logarit trong công thức này không thay đổi giá trị của 1 từ mà chỉ thu hẹp khoảng giá trị của từ đó. Việc thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF. Tuy nhiên, việc thay đổi khoảng giá trị sẽ giúp tỷ lệ giữa IDF và TF tương đồng để dùng cho công thức tính TF-IDF như sau:

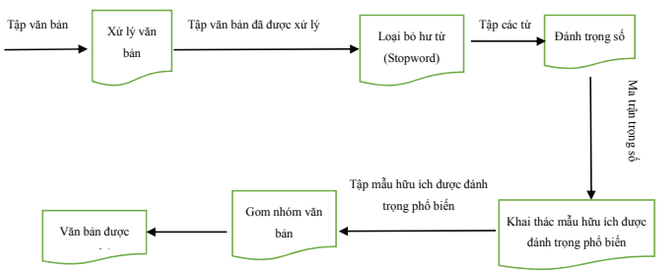
$$\text{TF-IDF}(t,d,D) = \text{tf}(t,\ d) \times \text{idf}\ (t,\ D) \quad (3)$$

Trong đó, những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này và ít xuất hiện trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản).

Nội dung nghiên cứu

Mô hình

Mô hình giải quyết bài toán gom nhóm văn bản tiếng Việt được thể hiện trên sơ đồ hình 1.



Hình 1. Mô hình giải quyết bài toán gom nhóm văn bản tiếng Việt.

Xử lý văn bản

Tách từ tiếng Việt: trước khi thực hiện gom nhóm văn bản cần tiến hành tách từ tiếng Việt. Không như tiếng Anh, ranh giới giữa các từ trong tiếng Việt không đơn thuần chỉ

là những khoảng trắng, đòi hỏi phải xử lý riêng trước. Ví dụ câu "Tổ quốc ta đẹp như tranh vẽ" sẽ được tách ra thành: Tổ_quốc | ta | đẹp | như | tranh_vẽ.

Nhận dạng tên riêng tiếng Việt (NER - Named Entity Recogniton)

Sau khi tách từ tiếng Việt, hệ thống sẽ tiến hành nhận dạng tên riêng tiếng Việt. Ví dụ câu "Triển lãm diễn ra từ 22 đến 30/6 tại Bảo tàng Mỹ thuật". Sau khi tách từ và tiến hành nhận dạng tên riêng tiếng Việt, sẽ ra thành các từ: Triển_lãm | diễn_ra | từ | 22/NUM | đến | 30/6/NUM | tại | Bảo_tàng Mỹ_thuật/PER |.

Để giải quyết bài toán tách từ và nhận dạng tên riêng tiếng Việt, chúng tôi sử dụng công cụ VnTokenizer để phân đoạn từ tiếng Việt và công cụ CLC_VN_NER để nhận dạng tên riêng tiếng Việt.

Loại bỏ các hư từ

Sau khi tách văn bản thành danh sách các từ, hệ thống tiến hành loại bỏ các hư từ (stopword) không có ý nghĩa đặc trưng (thì, là, nhỉ, vâng, nếu, như…). Để loại bỏ các hư từ, chúng tôi sử dụng từ điển các hư từ được chúng tôi tổng hợp lại (gồm 880 từ).

Thuật toán tìm mẫu hữu ích được đánh trọng phổ biến

Như đã được trình bày ở trên, các công trình trước đây quan tâm đến trọng số của các item hoặc vừa quan tâm đến trọng số của các item, lẫn trọng số của mỗi item trong các giao dịch. Đối với CSDL được xây dựng từ các văn bản, chỉ quan tâm đến trọng số của các item trong các giao dịch. Vì vậy, có thể áp dụng thuật toán MWIT-FWUI (hình 2) để khai thác mẫu hữu ích được đánh trọng phổ biến. Trong phần này, các tác giả đề xuất thuật toán cho phép khai thác mẫu hữu ích được đánh trọng phổ biến từ ma trận tần số xuất hiện của các từ trong văn bản như trên hình 3.

Định nghĩa 1: theo [9] Trọng số hữu ích của giao dịch t_k, ký hiệu twu (t_k), được tính theo công thức (4).

$$twu(t_k) = \frac{\sum_{i_j \in S(t_k)} w_j \times x_{ki_j}}{|t_k|} \quad (4)$$

Trong đó, x_{kij} là trọng số của item i_j trong giao dịch thứ t_k.

Định nghĩa 2 [9]: độ hỗ trợ hữu ích (wus - weighted utility support) của một itemset tính theo công thức (5).

$$wus(X) = \frac{\sum_{t_k \in t(X)} twu(t_k)}{\sum_{t_k \in T} twu(t_k)} \quad (5)$$

Thuật toán MWIT-FWUI

Input: CSDL *QD* chứa số lượng các từ trong văn bản và ngưỡng min_wus.

Output: Tập *W* chứa tất cả các mẫu hữu ích được đánh trọng phổ biến thỏa ngưỡng min_wus từ *QD*.

Phương pháp:

1. Từ *QD*, tính trọng số TF - IDF của mỗi từ trong văn bản // dùng công thức (1), (2)
2. Tính trọng số hữu ích giao dịch (*twu*) của mỗi văn bản // dùng công thức (4)
3. Sử dụng công thức (5) tính *wus* cho mỗi từ đơn trong các văn bản
4. $W = \emptyset$
5. P = Tập các từ đơn có *wus* thỏa *min_wus*
6. Gọi hàm FWUI_EXTEND (*P*)

FUI_EXTEND (*P*)

7. For all $n_i \in P$ do
8. Thêm $(n_i, wus(n_i))$ vào W
9. $P_i = \emptyset$
10. For all $n_j \in P$ with $j > i$ do
11. $X = n_i \cup n_j$
12. Tính độ hỗ trợ hữu ích wus(X) // dùng công thức (5)
13. Nếu (wus(X) ≥ min_wus) thì
14. Thêm X vào P_i
15. FWUI_EXTEND (P_i, min_wus)

Hình 2. Thuật toán MWIT-FWUI.

Thuật toán Gom nhóm văn bản

Input: Tập văn bản $D = \{d_1, d_2, ..., d_n\}$, mẫu hữu ích được đánh trọng phổ biến

Output: Tập văn bản đã được gom nhóm Clus = {clus1, clus2, ..., clusk}

Phương pháp:

Bước 1: Khởi tạo

Xây dựng ma trận tương đương A với a[i][j] = số mẫu phổ biến giống nhau giữa hai văn bản

Bước 2: Gom nhóm văn bản

1. Tìm giá trị nhỏ nhất (min) khác 0 trong ma trận A
2. Tìm giá trị lớn nhất (max) trong ma trận A và tìm tất cả các cặp văn bản chưa được gom nhóm trong ma trận A có giá trị = giá trị max (*)
3. While (True)

Nếu max = min thì

Tất cả các cặp văn bản chưa được gom nhóm sẽ được gom lại thành một nhóm mới

Ngược lại nếu max > min ta tiến hành theo trình tự sau:

1. Duyệt lần lượt các cặp văn bản tìm được, nếu có 2 văn bản trong cặp văn bản mà không thuộc vào nhóm đã có sẵn thì 2 văn bản này sẽ được sử dụng để gom thành 1 nhóm mới.

2. Duyệt lần lượt các cặp văn bản, nếu có 1 trong 2 văn bản trong cặp văn bản thuộc vào nhóm đã có sẵn thì văn bản còn lại sẽ được gom vào nhóm đã có này.

3. Gán giá trị cho các cặp văn bản = 0, tiếp tục quay trở lại thực hiện việc tìm giá trị max trong A(*).

Nếu còn bất kỳ văn bản nào chưa được gom nhóm thì mỗi văn bản này sẽ được sử dụng để gom lại thành 1 nhóm mới.

End while.

Kết thúc việc gom nhóm.

Hình 3. Thuật toán Gom nhóm văn bản.

Ví dụ minh họa

Xét cơ sở dữ liệu tần số xuất hiện của các từ như trên bảng 1.

Bảng 1. Cơ sở dữ liệu số lần xuất hiện của các từ.

TID \ ITEM	A	B	C	D	E
1	2	0	3	0	4
2	0	2	0	1	3
3	0	4	2	2	1
4	3	0	1	2	2
5	0	3	0	0	3
6	1	0	2	3	0
7	2	2	3	0	0
8	0	2	2	0	0
9	0	0	4	3	2

Ta có: tập văn bản $D = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9\}$, và tập các từ $I = \{A, B, C, D, E\}$, với $d_1 = \{2, 0, 3, 0, 4\}$, nghĩa là trong văn bản d_1 có 2 từ A, 3 từ C, 4 từ E, không có từ B, D.

Khai thác mẫu hữu ích được đánh trọng phổ biến:

Sử dụng công thức (1) để tính tần số xuất hiện TF của các từ trong văn bản. Vì item A xuất hiện ở văn bản 1, 4, 6, 7, nên TF(A) được tính như sau: TF $(A, d_1) = \frac{2}{9}$; TF $(A, d_4) = \frac{3}{8}$; TF $(A, d_6) = \frac{1}{6}$; TF $(A, d_7) = \frac{2}{7}$. Tương tự cho các item còn lại, thu được kết quả như ở bảng 2.

Bảng 2. Tần số xuất hiện của các từ trong văn bản.

TID \ ITEM	A	B	C	D	E
1	0,222	0	0,333	0	0,444
2	0	0,333	0	0,167	0,5
3	0	0,444	0,222	0,222	0,111
4	0,375	0	0,125	0,25	0,25
5	0	0,5	0	0	0,5
6	0,167	0	0,333	0,5	0
7	0,286	0,286	0,429	0	0
8	0	0,5	0,5	0	0
9	0	0	0,444	0,333	0,222

Sử dụng công thức (2) để tính trọng số của các từ trong văn bản. Mỗi từ chỉ có 1 giá trị IDF duy nhất trong tập văn bản, được xem như là trọng số của mỗi từ trong tập văn bản. Ta có: IDF (A, D) = $\log \frac{9}{4}$; tương tự thu được kết quả như ở bảng 3.

Bảng 3. Trọng số của các Item.

Item	IDF
A	0,35
B	0,25
C	0,11
D	0,25
E	0,18

Tính trọng số hữu ích giao dịch (twu) của mỗi văn bản: từ dữ liệu ở bảng 2 và bảng 3, tính được trọng số hữu ích giao dịch (twu) của mỗi văn bản theo công thức (4):

$$twu(d_1) = \frac{0,222\times0,35+0,333\times0,11+0,444\times0,18}{3} = 0,064$$

Tương tự, ta được kết quả như trên bảng 4.

Bảng 4. Trọng số hữu ích giao dịch của mỗi văn bản.

Tids	twu
1	0,064
2	0,072
3	0,0535
4	0,063
5	0,108
6	0,074
7	0,073
8	0,0915
9	0,058
SUM	**0,657**

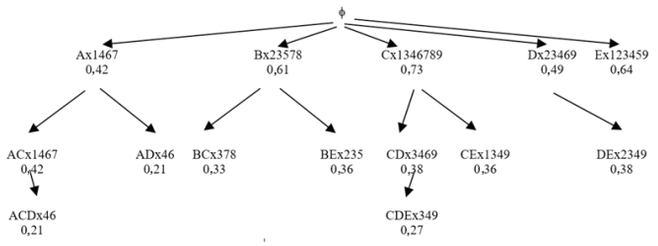
Tính độ hỗ trợ hữu ích (wus): từ dữ liệu ở bảng 2 và 4, ta tính độ hỗ trợ hữu ích theo công thức (5), trong đó lấy min_wus = 0,2. Item A xuất hiện ở văn bản 1, 4, 6, 7 nên:

$$wus\ (A) = \frac{0,064+0,063+0,074+0,073}{0,657} = 0,42$$

Tương tự, ta được wus (B) = 0,61, wus (C) = 0,73, wus (D) = 0,49, wus (E) = 0,64. Tất cả các giá trị này đều thỏa min_wus = 0,2, do đó tất cả các item đơn này sẽ được thêm vào cây mẫu phổ biến: $\phi = \{A, B, C, D, E\}$.

Xét lớp tương đương A, A kết hợp với B, ta được một itemset mới AB với wus (AB) = 0,12 < min_wus, vì vậy AB không được thêm vào A. Tương tự, wus (AC) = 0,42 ≥ min_wus, thêm AC vào A, ta được A = {AC}; wus (AD) = 0,21 ≥ min_wus, thêm AD vào A, ta được A = {AC, AD}; wus (AE) = 0,19 < min_wus, vì vậy AE không được thêm vào A.

Sau khi thực hiện xong ở lớp A, sẽ gọi thuật toán đệ quy để tiếp tục tạo ra các lớp tương đương sau nó. Thực hiện tương tự với các lớp B, C, D, E, cuối cùng ta có được mẫu hữu ích được đánh trọng phổ biến thỏa mãn min_wus = 0,2 là {A, B, C, D, E, AC, AD, BC, BE, CD, CE, DE, ACD, CDE} trên hình 4.



Hình 4. Cây mẫu hữu ích được đánh trọng phổ biến với min_wus = 0,2.

Gom nhóm văn bản:

- Xây dựng ma trận tương đương: từ dữ liệu ở bảng 2 và hình 4, ta xây dựng được ma trận tương đương với a[i][j] = số mẫu phổ biến giống nhau giữa hai văn bản (bảng 5).

Bảng 5. Ma trận tương đương.

	1	2	3	4	5	6	7	8	9
1		1	3	5	1	3	3	1	3
2			5	3	3	1	1	1	3
3				7	3	3	3	3	7
4					1	7	3	1	7
5						0	1	1	1
6							3	1	3
7								3	1
8									1

- Thực hiện gom nhóm (5 bước):

Bước 1: tìm giá trị nhỏ nhất khác 0 trong ma trận tương đương A: min = 1.

Bước 2: tìm giá trị lớn nhất trong A: max = 7.

Bước 3: tìm các cặp văn bản có giá trị bằng max. Ta thấy (3, 4); (3, 9); (4, 6); (4, 9) có giá trị bằng 7; gom (3, 4, 6, 9) là 1 nhóm và gán giá trị cho các cặp văn bản vừa tìm được bằng 0; còn lại các văn bản (1, 2, 5, 7, 8) chưa được gom nhóm (bảng 6).

Bảng 6. Ma trận tương đương với các văn bản (3, 4, 6, 9) đã được gom nhóm.

	1	2	3	4	5	6	7	8	9
1		1	3	5	1	3	3	1	3
2			5	3	3	1	1	1	3
3				0	3	3	3	3	0
4					1	0	3	1	0
5						0	1	1	1
6							3	1	3
7								3	1
8									1

Bước 4: tìm giá trị max trong A và các cặp văn bản có cùng giá trị với max. Ta thấy max = 5; các cặp văn bản (1, 4); (2, 3) có cùng giá trị với max; (1, 4); (2, 3) có văn bản 3, 4 thuộc nhóm (3, 4, 6, 9), do đó ta gom 1, 2 vào nhóm này. Thu được (1, 2, 3, 4, 6, 9) là 1 nhóm và gán giá trị cho các cặp văn bản vừa tìm được bằng 0; còn lại văn bản (5, 7, 8) chưa được gom nhóm (bảng 7).

Bảng 7. Ma trận tương đương với các văn bản (1, 2, 3, 4, 6, 9) đã được gom nhóm.

	1	2	3	4	5	6	7	8	9
1		1	3	0	1	3	3	1	3
2			0	3	3	1	1	1	3
3				0	3	3	3	3	0
4					1	0	3	1	0
5						0	1	1	1
6							3	1	3
7								3	1
8									1

Bước 5: tiếp tục tìm giá trị max trong A và các cặp văn bản có cùng giá trị với max. Ta thấy max = 3, các cặp văn bản (1, 3); (1, 6); (1, 7); (1, 9); (2, 4); (2, 5); (2, 9); (3, 5); (3, 6); (3,7); (3, 8); (4, 7); (6, 7); (6, 9); (7, 8) có cùng giá trị với max. Trong các cặp văn bản trên, có cặp văn bản (7, 8) chưa thuộc nhóm văn bản đã được gom cụm, do đó ta gom (7, 8) thành 1 nhóm và gán giá trị cho (7, 8) bằng 0.

Xét tiếp (2, 5) có văn bản 2 thuộc nhóm (1, 2, 3, 4, 6, 9), do đó ta gom 5 vào nhóm đã có này và gán giá trị (2, 5) bằng 0, được (1, 2, 3, 4, 5, 6, 9) là 1 nhóm; (7, 8) là 1 nhóm, gán giá trị cho các cặp văn bản vừa tìm được bằng 0.

Như vậy, tất cả văn bản đã được gom nhóm; kết thúc việc gom nhóm.

Từ 9 văn bản đã cho ban đầu ta gom được thành 2 nhóm: nhóm 1 gồm các văn bản (1, 2, 3, 4, 5, 6, 9) và nhóm 2 gồm các văn bản (7, 8) như trên bảng 8.

Bảng 8. Ma trận tương đương với tất các văn bản đã được gom nhóm.

	1	2	3	4	5	6	7	8	9
1		1	0	0	1	0	0	1	0
2			0	0	0	1	1	1	0
3				0	0	0	0	0	0
4					1	0	0	1	0
5						0	1	1	1
6							0	1	0
7								0	1
8									1

Kết quả thực nghiệm và đánh giá

Các thuật toán được xây dựng trên nền .NET Framework 4.0, sử dụng Visual Studio 2010. Kho ngữ liệu thử nghiệm được chúng tôi rút trích tự động kết hợp với định dạng thủ công bao gồm 1.600 văn bản thuộc 16 chủ đề (bảng 9) từ các diễn đàn trực tuyến như "www.vnexpress.net", "www.dantri.com.vn", "www.thanhnien.vn".

Bảng 9. Tập dữ liệu thực nghiệm.

STT	Chủ đề	Số lượng văn bản	Bộ dữ liệu thử nghiệm
1	Đời sống	100	1
2	Mỹ thuật	100	
3	Virus-Hacker	100	
4	Thể thao	100	
5	Tennis	100	2
6	Y khoa	100	
7	Âm nhạc	100	
8	Sân khấu điện ảnh	100	
9	Công nghệ thông tin	100	
10	Thời trang	100	3
11	Bóng đá	100	
12	Bất động sản	100	
13	Ẩm thực	100	
14	Bệnh cúm	100	
15	Xe	100	
16	Hình sự	100	
TC	16	1.600	

Sử dụng độ đo F (F-measure) [4] để đánh giá tính hiệu quả của thuật toán gom nhóm dùng mẫu phổ biến. Độ đo F được tính theo công thức từ (6-9):

$$F(i,j) = \frac{2 \times P(i,j) \times R(i,j)}{P(i,j) + R(i,j)} \quad (6)$$

$$F = \sum_i \frac{n_i}{n} \max_j F(i,j) \quad (7)$$

$$P(i,j) = \frac{n_{ij}}{n_j} \quad (8)$$

$$R(i,j) = \frac{n_{ij}}{n_i} \quad (9)$$

Trong đó: n_i là số văn bản của lớp i, n_j là số văn bản của cụm j, n_{ij} là số văn bản của lớp i trong cụm j, n là tổng số văn bản trong bộ dữ liệu; P(i, j) là độ rõ (Precision) của cụm j trong lớp i; R(i, j) là độ bao phủ (Recall) của lớp i trong cụm j; F(i, j) là độ đo F của cụm j trong lớp i. Độ đo F càng cao, kết quả gom nhóm càng tốt.

Tiến hành so sánh với phương pháp gom nhóm văn bản dựa trên mẫu phổ biến [7] với 1.600 văn bản trên 3 bộ dữ liệu. Bộ dữ liệu 1 gồm 4 chủ đề (Đời sống, Mỹ thuật, Virus-Hacker, Thể thao) với 400 văn bản. Kết quả độ đo F với các ngưỡng tương ứng như trong bảng 10.

Bảng 10. Kết quả độ đo F của bộ dữ liệu 1.

F \ MS	0,15	0,14	0,13	0,12	0,11	0,1	0,09	0,085	0,08	0,07	0,05
MC	0,57	0,59	0,52	0,56	0,64	0,6	0,64	0,61	0,62	0,63	0,68
MC-W	0,7	0,76	0,78	0,72	0,82	0,8	0,74	0,82	0,83	0,84	0,83

Đối với bộ dữ liệu 1, kết quả gom nhóm đạt được khá cao. Ở bảng 10 ta thấy, với thuật toán MC, độ đo F đạt kết quả cao nhất là 0,68 (khoảng 68%) số văn bản được gom vào đúng nhóm. Với thuật toán MC-W, độ đo F đạt kết quả cao nhất là 0,84 (khoảng 84%) số văn bản được gom vào đúng nhóm.

Bộ dữ liệu 2 gồm 5 chủ đề (Tennis, Y khoa, Âm nhạc, Sân khấu điện ảnh, Công nghệ thông tin) với 500 văn bản. Kết quả chạy được với các ngưỡng tương ứng như trong bảng 11.

Bảng 11. Kết quả độ đo F của bộ dữ liệu 2.

F \ MS	0,15	0,14	0,13	0,12	0,11	0,1	0,095	0,09	0,085	0,08	0,07
MC	0,46	0,4	0,51	0,5	0,42	0,41	0,4	0,41	0,49	0,42	0,47
MC-W	0,48	0,47	0,51	0,5	0,48	0,48	0,49	0,49	0,49	0,48	0,5

Đối với bộ dữ liệu 2, bảng 11 cho thấy thuật toán MC và thuật toán MC-W đều có độ đo F cao nhất là 0,51 (51%), số văn bản được gom vào đúng nhóm.

Bộ dữ liệu 3 gồm 7 chủ đề (Thời trang, Bóng đá, Bất động sản, Ẩm thực, Bệnh cúm, Xe, Hình sự) với 700 văn bản. Kết quả chạy được với các ngưỡng tương ứng như trong bảng 12.

Bảng 12. Kết quả độ đo F của bộ dữ liệu 3.

F \ MS	0,1	0,095	0,09	0,085	0,08	0,075	0,07	0,065	0,06	0,05	0,045	0,04	0,035
MC	0,37	0,4	0,37	0,37	0,37	0,38	0,44	0,48	0,52	0,5	0,49	0,5	0,54
MC-W	0,43	0,46	0,54	0,52	0,56	0,57	0,53	0,51	0,53	0,57	0,54	0,54	0,59

Đối với bộ dữ liệu 3 ở bảng 12 cho thấy, với thuật toán MC, độ đo F đạt kết quả cao nhất là 0,54 (khoảng 54%) số văn bản được gom vào đúng nhóm. Với thuật toán MC-W, độ đo F đạt kết quả cao nhất là 0,59 (khoảng 59%) số văn bản được gom vào đúng nhóm.

Từ kết quả thử nghiệm cho thấy, thuật toán MC-W gom nhóm văn bản tốt hơn và có độ chính xác cao hơn so với thuật toán MC. Với mỗi ngưỡng khác nhau sẽ cho kết quả độ đo F khác nhau. Do đó, việc xác định ngưỡng phù hợp cho bài toán vẫn là một thách thức với bài toán tìm mẫu phổ biến nói chung, bài toán gom nhóm văn bản sử dụng mẫu phổ biến nói riêng.

Kết luận

Bài báo đề xuất một phương pháp mới để gom nhóm văn bản dựa vào mẫu hữu ích được đánh trọng phổ biến. Đầu tiên, chúng tôi tính TF-IDF cho mỗi từ trong văn bản nhằm tạo ra một ma trận trọng số cho tập văn bản, sau đó sử dụng thuật toán MWIT-FWUI để khai thác các mẫu hữu ích phổ biến từ ma trận trọng số. Cuối cùng, dựa vào tập các mẫu hữu ích khai thác được để tiến hành gom nhóm văn bản theo thuật toán MC. Với phương pháp này, hệ thống đã cải thiện được đáng kể độ chính xác của việc gom nhóm văn bản, cụ thể là độ đo F so với phương pháp gom nhóm dựa vào mẫu phổ biến.

Trong tương lai, các tác giả sẽ tiếp tục áp dụng thêm các đặc trưng ngôn ngữ như từ nội dung, tên riêng… vào gom nhóm văn bản. Mặc khác, chúng tôi tiếp tục thử nghiệm sử dụng phương pháp đánh trọng cho các bài toán khác như: Tìm kiếm văn bản, tóm tắt văn bản, phân loại văn bản…

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Quỹ Phát triển KH&CN Quốc gia (NAFOSTED) trong khuôn khổ đề tài mã số 102.05-2015.10. Các tác giả xin trân trọng cảm ơn.

TÀI LIỆU THAM KHẢO

[1] Đỗ Phúc, Mai Xuân Hùng, Nguyễn Thị Kim Phụng (2008), "Gom cụm đồ thị và ứng dụng vào việc rút trích nội dung chính của khối thông điệp trên diễn đàn thảo luận", *Tạp chí Phát triển khoa học và c*ông nghệ, **11(5)**, tr.21-32.

[2] Dương Thị Thùy Vân (2010), "Gom cụm tài liệu dựa trên thực thể có tên", *Tạp chí Khoa học ứng dụng*, *Trường Đại học Tôn Đức Thắng,* **11**, tr.42-45.

[3] F. Beil, M. Ester, X. Xu (2002), "Frequent term-based text clustering", *Proceedings of the 8th ACM SIGKDD*, pp.436-442.

[4] M. Steinbach, G. Karypis, V. Kumar (2000), "A comparison of document clustering techniques", *KDD-2000 Workshop on Text Mining*, pp.109-110.

[5] H. Edith, A. Rene, J. Carrasco-Ochoa, J. Martinez-Trinidad (2006), "Document clustering based on maximal frequent sequences", *Proceedings of the FinTAL*, *LNAI,* **4139**, pp.257-267.

[6] B. Fung, K. Wang, M. Ester (2003), "Hierarchical document clustering using frequent itemsets", *Proceedings of the 3rd SIAM International Conference on Data Mining*, pp.59-70.

[7] Y. Li, S. Chung, J. Holt (2008), "Text document clustering based on frequent word meaning sequences", *Data & Knowledge Engineering*, **64(1)**, pp.381-404.

[8] W. Zhang, T. Yoshida, X. Tang, Q. Wang (2010), "Text clustering using frequent itemsets", *Knowledge-Based Systems*, **23(5)**, pp.379-388.

[9] B. Vo, B. Le, J. Jung (2012), "A tree-based approach for mining frequent weighted utility itemsets", *ICCCI'12, LNAI Springer*, **7653**,