ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

PHÂN TÍCH CẢM XÚC TRONG VĂN BẢN Y KHOA

HỘI ĐỒNG: KHOA HỌC MÁY TÍNH

Giáo viên hướng dẫn:
GS. TS. Cao Hoàng Trụ

Giáo viên phản biện:
GS. TS. Phan Thị Tươi

Sinh viên thực hiện:
Nguyễn Đức Trí (51204052)
Nguyễn Diệp Phương Linh (51201899)

Thành phố Hồ Chí Minh, 12/2016

Lời cam đoan

Lời cảm ơn

Tóm tắt luận văn

Mục lục

1 Giới thiệu **7**
1.1 Động cơ thực hiện đề tài . . . 7
1.2 Phạm vi và đóng góp của luận văn . . . 7
1.3 Cấu trúc luận văn . . . 7

2 Các công trình liên quan **8**

3 Kiến thức nền tảng **9**
3.1 Phân tích cảm xúc . . . 9
3.2 Phân tích phủ định . . . 9
3.3 Phương pháp học máy Support Vector Machine . . . 11
3.4 Phương pháp đánh giá độ đồng nhất Cohen's kappa . . . 14
3.5 Các thư viện và công cụ hỗ trợ . . . 16

4 Phương pháp đề xuất **19**
4.1 Mô tả bài toán . . . 19
4.2 Kiến trúc tổng quan . . . 19
4.3 N-gram . . . 21
4.4 Change Phrase . . . 24
4.5 Thành phần phủ định . . . 26
4.6 Đặc trưng mở rộng SO-CAL . . . 28

5 Hiện thực hệ thống **32**
5.1 Thư viện và công cụ sử dụng . . . 32
5.2 Hiện thực rút trích đặc trưng . . . 34
5.3 Hiện thực bộ phân loại SVM . . . 37

6 Thí nghiệm và đánh giá **39**
6.1 Thu thập và đánh giá dữ liệu . . . 39
6.2 Phương pháp đánh giá . . . 42
6.3 Kết quả thí nghiệm . . . 45
6.4 Các phân tích mở rộng . . . 48

7 Tổng kết **49**
7.1 Kết quả đạt được . . . 49
7.2 Hạn chế và hướng phát triển . . . 49

Danh sách hình vẽ

1 Minh họa mô hình phân loại dữ liệu có nhãn 12
2 Caption for LOF 12
3 Caption for LOF 13
4 Minh họa phương pháp soft-margin 14
5 Kiến trúc tổng quát của MetaMap [2] 17
6 Ví dụ kết quả chạy MetaMap 18
7 Kiến trúc tổng quan xây dựng hệ thống 20
8 MetaMap sử dụng nguồn tài nguyên UMLS, giúp tra cứu tên kiểu của 1 thuật ngữ y học 23
9 Giải thuật trích xuất đặc trưng n-gram 23
10 Giao diện web tương tác trực tiếp của MetaMap 33
11 Batch MetaMap 34
12 Công cụ online để tính điểm SO-CAL 36
13 Kết hợp các đặc trưng trước khi đưa vào SVM huấn luyện 37
14 Tóm tắt của 1 bài báo 39
15 Mô hình thực thể liên kết tăng cường của cơ sở dữ liệu 40
16 Giao diện trang đánh nhãn dữ liệu 40
17 Thông tin một bản ghi thuộc bảng `Sentence` 41
18 Nhóm nút chức năng hỗ trợ người dùng lựa chọn phân loại 41
19 Quy trình xử lý gắn nhãn dữ liệu của trang web 42
20 Các thành phần trong các phép đo Độ chính xác, Độ bao phủ và f1 43
21 Mối quan hệ giữa tham số min_df, cách vector hóa và độ đo F 46
22 Kết hợp các n-gram 47

Danh sách bảng

1 Minh họa phương pháp phân loại OvA cho bài toán phân tích cảm xúc trong bệnh án điện tử 14
2 Thống kê các câu được đánh nhãn 15
3 Thang đo đánh giá độ đồng nhất dựa trên giá trị κ 16
4 Các đặc trưng *Change phrase* 25
5 Tỉ lệ tác động của một số từ 31
6 Một số mẫu từ tập dữ liệu sau khi thu thập 39
7 Một số mẫu từ tập dữ liệu sau khi đánh nhãn 41
8 Các thử nghiệm nhằm tối ưu hóa đặc trưng n-gram 45
9 Các thử nghiệm nhằm tối ưu đặc trưng Phủ định 47
10 Các thử nghiệm kết hợp các đặc trưng cơ bản 48
11 Các thử nghiệm kết hợp các đặc trưng cơ bản với đặc trưng mở rộng SO-CAL 48

1 Giới thiệu

1.1 Động cơ thực hiện đề tài

1.2 Phạm vi và đóng góp của luận văn

1.3 Cấu trúc luận văn

2 Các công trình liên quan

[1]

3 Kiến thức nền tảng

Trong phần này chúng tôi sẽ trình bày khái niệm cơ bản về lĩnh vực Phân tích cảm xúc trong văn bản nói chung và trong báo cáo y khoa nói riêng, cùng với nền tảng kiến thức của một số khái niệm và phương pháp sử dụng trong quá trình phân tích cảm xúc.

3.1 Phân tích cảm xúc

Phân tích cảm xúc trong văn bản nói chung

Phân tích cảm xúc trong báo cáo y khoa

3.2 Phân tích phủ định

Theo bài báo [7], xấp xỉ một nửa số câu mô tả trong các văn bản lâm sàng và báo cáo y khoa chịu sự can thiệp của các yếu tố phủ định. Việc xuất hiện thành phần phủ định trong câu có thể dẫn đến thay đổi hoàn toàn tính phân cực của câu. Vì vậy hiện thực tốt bước tự động phân tích phủ định góp phần quan trọng để nâng cao hiệu quả phân loại tính phân cực của câu trong văn bản y khoa.

Do đó, bài toán phân tích phủ định được đặt ra nhằm xác định thành phần phủ định và phạm vi chịu sự phủ định trong câu, từ đó phân tích ảnh hưởng của các yếu tố phủ định lên tính phân cực của câu. Nhiều thuật toán phân tích phủ định đã được hiện thực trên văn bản tiếng Anh [**Aronow1999**, 7, 20, 13, 35], và một số trong đó được phát triển để nhận diện phủ định cho các ngôn ngữ khác [3, 9, 8, 11, 15]. Ở phạm vi báo cáo luận văn này, chúng tôi xem bước phân tích phủ định như một bài toán con trong bài toán phân tích cảm xúc chung. Để giải quyết bài toán này, trước hết cần hiểu rõ các thành phần cơ bản của sự phủ định và hình thức tồn tại sự phủ định trong câu.

Thành phần phủ định

Thành phần phủ định (*negation*) là một khái niệm chỉ một từ hoặc cụm từ mang ý nghĩa phủ nhận sự tồn tại của một yếu tố khác [29]. Ở ví dụ 1, từ `no` - đóng vai trò là thành phần phủ định - phủ nhận sự tồn tại của cụm từ `significant effect`, hay nói cách khác, cụm danh từ `significant effect` chịu ảnh hưởng của yếu tố phủ định trong câu.

Ví dụ 1

```
Early administration of oral steroid medication in
patients with acute sciatica had ([no] significant ef-
fect).
```

Hình thức phủ định

Trong ngữ pháp tiếng Anh [16], phủ định có thể xảy ra theo hai hình thức: phủ định hình thái (*morphological negation*) và phủ định cú pháp (*syntactic negation*). Trong đó, phủ định hình thái được tạo ra khi thay đổi từ gốc bằng những tiền tố phủ định (như "dis-", "non-", "un-") hoặc hậu tố phủ định (như "-less"), còn phủ định cú pháp là hình thức phủ định sử dụng từ ngữ phủ định hoặc mẫu cú pháp riêng biệt rõ ràng và mang ý nghĩa phủ nhận một từ hoặc cụm từ khác trong cùng câu hoặc ở câu liên quan.

Phạm vi của phủ định hình thái chỉ giới hạn ở một từ riêng lẻ nên không tác động lên yếu tố khác trong câu. Vì vậy bài toán phân tích phủ định chủ yếu tập trung phân tích dạng phủ định cú pháp, bao gồm hai thành phần chính là thành phần phủ định (có thể bao gồm một từ hoặc một cụm từ, gọi chung là từ phủ định) và phạm vi phủ định của từ đó. Ví dụ 1 là một trường hợp đơn giản nhất của phủ định cú pháp.

Từ phủ định

Trên thực tế, bài toán phân tích phủ định thường gặp khó khăn bởi sự đa dạng về từ phủ định cũng như vị trí tương đối giữa từ phủ định và từ bị phủ định trong câu. Ở ví dụ 2, từ phủ định không chỉ là những từ đơn giản như "no", "not" mà còn bao gồm từ phủ định khác như "without", "rule out", "exclude", ...
Ví dụ 2

```
Mildly hyperinflated lungs ([without] focal opacity).
(Myelomeningocele is [excluded]).
```

Bên cạnh đó, những động từ như "rule out", "exclude" mang ý nghĩa khác nhau khi xuất hiện trong những trường hợp đặc biệt. Ví dụ 3 minh họa câu mệnh lệnh yêu cầu khám trong ghi chú của bác sĩ, cho thấy khả năng viêm phổi (*pneumonia*) vẫn còn hiện diện. Vì thế "rule out" trong câu này không thể hiện sự phủ định.
Ví dụ 3

```
(<Rule out> pneumonia).
```

Tuy nhiên, khi được tìm thấy trong câu bị động như ở ví dụ 4, nó thể hiện sự phủ định rõ ràng khi phủ nhận khả năng bị ung thư phổi.
Ví dụ 4

```
(The possibility of lung cancer is [ruled out]).
```

Mặt khác, nếu dạng bị động của những động từ này bị phủ định, sự phủ định sẽ bị loại bỏ như ở ví dụ 5.
Ví dụ 5

```
(It is <not ruled out> that the ureterocele opens into
the vagina).
```

Bởi sự phức tạp trong việc nhận diện ý nghĩa phủ định của các từ phủ định nên cần thiết có một danh sách các từ phủ định được lọc và phân loại rõ ràng. Giải thuật phủ định NegEx (sẽ được đề cập rõ hơn ở chương 4) đã xây dựng một danh sách thuật ngữ phủ định[1] chia làm 3 loại:

- Phủ định tiền điều kiện (*pre-condition negation term*) bao gồm những từ phủ định có vị trí đứng trước những cụm từ bị nó phủ định trong câu. Ví dụ như `without`, `absence of`, `rule out`...
- Phủ định hậu điều kiện (*post-condition negation term*) bao gồm những từ phủ định có vị trí đứng sau những cụm từ bị nó phủ định trong câu và thường ở thể bị động. Ví dụ như `be ruled out`, ...

[1]https://code.google.com/archive/p/negex/wikis/NegExTerms.wiki

- Giả phủ định (*pseudo negation term*) bao gồm những cụm từ trông có vẻ như từ phủ định nhưng không mang ý nghĩa phủ định. Ví dụ như `not certain if`, `without difficulty`...

Việc phân loại các từ phủ định như trên giúp trả lời hai câu hỏi: từ nào trong câu là từ có mang ý nghĩa phủ định và vị trí của từ bị phủ định là trước hay sau từ phủ định đó. Vấn đề còn lại là xác định tầm vực ảnh hưởng của từ phủ định trong câu.

Phạm vi phủ định

Xét về tầm vực phủ định, phủ định cú pháp được chia hai loại là phủ định liên câu (*intersentential negation*) và phủ định trong câu (*sentential negation*) [10]. Khác với phủ định liên câu - dạng phủ định mà từ phủ định có ảnh hưởng phủ định lên câu khác, phủ định trong câu có từ phủ định và từ bị phủ định cùng tồn tại trong một câu (ví dụ 6). Với đề tài luận văn này, chúng tôi chỉ xem xét đến dạng phủ định trong câu.
Ví dụ 6

> Phủ định liên câu: `Is this treatment effective? [No].`
> Phủ định trong câu: `The treatment does [not] reveal the etiology of the patient's pain.`

Để giải quyết vấn đề xác định phạm vi phủ định trong câu cần xây dựng một danh sách chứa các thuật ngữ kết thúc (*termination terms*)[1]. Danh sách này gồm những từ báo hiệu kết thúc sự ảnh hưởng của từ phủ định lên các thành phần không liên quan trong câu. Ở ví dụ 7, từ `but` báo hiệu kết thúc phạm vi phủ định gây ra bởi từ phủ định `denies`.
Ví dụ 7

> `Patient ([denies] chest pain) but continues to experience SOB.`

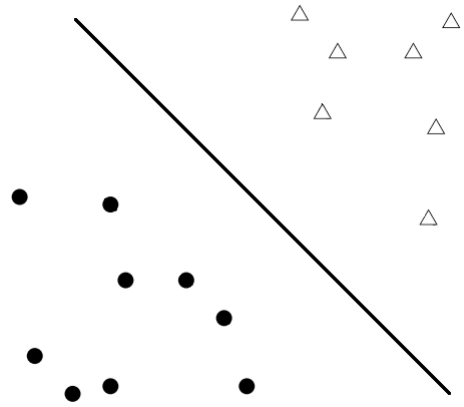
Nếu một từ (hoặc cụm từ) được hỏi nằm trong phạm vi phủ định thì từ đó bị phủ định. Ở ví dụ 7, từ `chest pain` bị phủ định vì nằm trong vùng phủ định của từ `denies`.

3.3 Phương pháp học máy Support Vector Machine

Với dữ liệu dạng văn bản, nhiều phương pháp phân loại có thể được sử dụng như *Support Vector Machine* (SVM), *Naive Bayes*, *Expectation–maximization algorithm*, ...[19]. Trong quá trình nghiên cứu và hiện thực, nhóm chọn mô hình SVM làm giải thuật nền tảng để dán nhãn các lớp cho tập dữ liệu bởi hiệu suất cao của phương pháp này trong việc phân loại nhãn văn bản [17].

SVM được Vapnik lần đầu tiên giới thiệu vào năm 1992 và từ đó trở thành một trong những giải thuật học máy được sử dụng phổ biến nhất bởi hiệu suất phân loại tốt trên những tập dữ liệu có kích thước không quá lớn. SVM được sử dụng chủ yếu để giải quyết các bài toán phân loại và bài toán phân tích hồi quy [6][19].

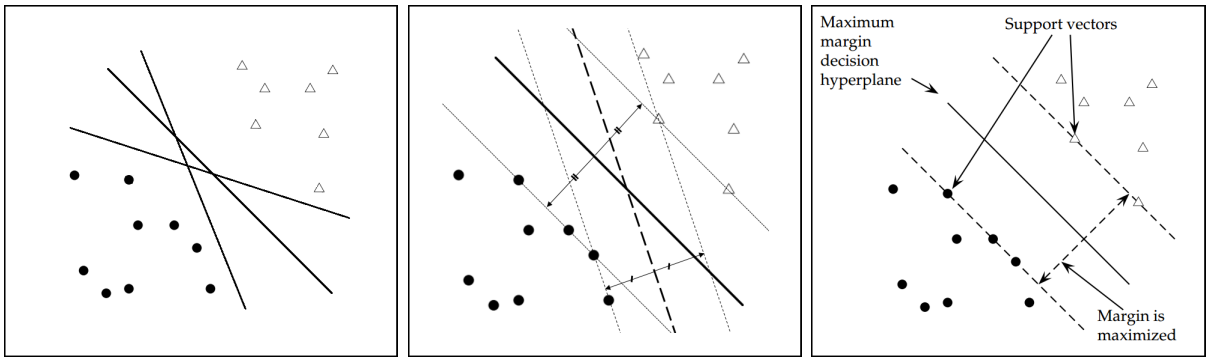
[1] https://code.google.com/archive/p/negex/wikis/NegExTerms.wiki



Hình 1: Minh họa mô hình phân loại dữ liệu có nhãn

Mô hình SVM chuẩn là một mô hình học máy có giám sát sử dụng thuật toán phân loại nhị phân. SVM nhận đầu vào là một tập dữ liệu biết trước nhãn của mỗi phần tử thuộc tập dữ liệu đó. Ví dụ như ở Hình 1, tập dữ liệu đầu vào gồm tập hợp các điểm biết trước phân lớp (hình tam giác, hình tròn). Nhiệm vụ của SVM là xây dựng một đường phân cách tuyến tính chia tập dữ liệu thành hai nhóm điểm thuộc hai lớp khác nhau, sao cho khi có một điểm mới xuất hiện chưa biết trước nhãn, từ vị trí của điểm này so với đường phân cách có thể dự đoán điểm mới thuộc nhóm phân lớp nào. Để giải quyết bài toán này, mô hình SVM sử dụng giải thuật tối ưu hóa khoảng cách giữa đường phân chia tuyến tính đến điểm dữ liệu gần nhất ở cả hai lớp.

Giải thuật tối ưu hóa khoảng cách



(A) (B) (C)

Hình 2: Minh họa giải thuật tối ưu hóa khoảng cách của mô hình SVM [19]

Với không gian dữ liệu tương đối đơn giản như Hình 2a, vấn đề đặt ra là có vô số đường tuyến tính có khả năng phân chia tập dữ liệu thành hai lớp phân biệt. Trong tập hợp các đường phân cách này, ta cần lựa chọn một đường tối ưu để tăng hiệu quả phân loại và giảm nhiễu cho tập dữ liệu bằng giải thuật tối ưu hóa khoảng cách của SVM.

Đầu tiên, đường phân cách cần nằm cách đều hai nhóm phân loại để đảm bảo xác suất phân loại điểm mới công bằng cho cả hai lớp (Hình 2b). Điều này có nghĩa là khoảng cách D từ đường phân cách đến điểm gần nhất ở cả hai lớp phải bằng nhau. Tuy nhiên, số lượng đường tuyến tính đảm bảo yêu cầu trên là vô số. Lúc này ta quan tâm đến độ

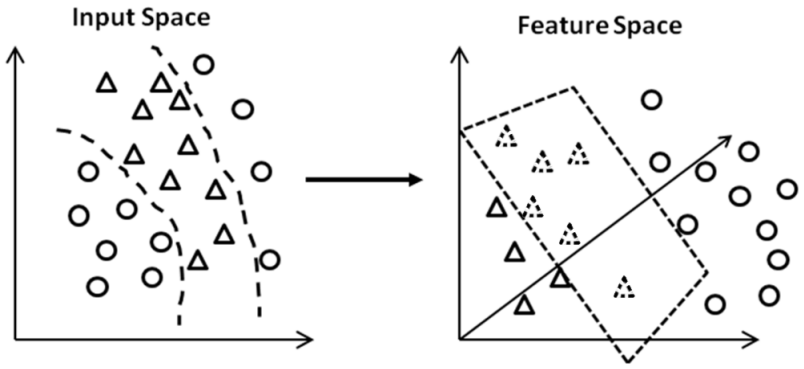
lớn của D: nếu D nhỏ, xác suất phân loại sai sẽ cao hơn do sai số của dữ liệu đầu vào trên thực tế. Vì vậy mô hình SVM chọn đường phân loại có khoảng cách lớn nhất đến điểm gần nhất ở cả hai lớp (Hình 2c). Các điểm dữ liệu thuộc hai lớp có vị trí gần nhất với đường thẳng phân loại gọi là *support vector*.

Kĩ thuật Kernel

Dựa vào đặc trưng về vị trí điểm dữ liệu, tập dữ liệu đầu vào có thể được chia làm hai loại:

- Khả phân cách tuyến tính: tồn tại ít nhất một đường thẳng thuộc không gian dữ liệu có thể phân chia tập dữ liệu xác định thành hai nhóm có nhãn khác nhau như Hình 1.
- Không khả phân cách tuyến tính: không tồn tại đường thẳng thuộc không gian tập dữ liệu có khả năng chia các điểm dữ liệu thành hai nhóm có nhãn khác nhau. Lúc này bài toán trở thành phân loại phi tuyến.

Trong trường hợp dữ liệu đầu vào không khả phân cách tuyến tính, cách giải quyết đầu tiên là biến đổi không gian dữ liệu trở thành khả phân cách tuyến tính. Nói cách khác, ta cần tìm một hàm ánh xạ sao cho với không gian dữ liệu sau khi ánh xạ tồn tại ít nhất một đường thẳng hoặc mặt phẳng tuyến tính có thể phân loại dữ liệu thành hai lớp. Để làm việc này ta có thể chỉnh sửa các đặc trưng của dữ liệu, hoặc suy diễn đặc trưng mới trên cơ sở những đặc trưng có sẵn. Ngoài ra, ta cũng có thể ánh xạ các điểm dữ liệu vào không gian có số chiều lớn hơn sao cho trong không gian đó có ít nhất một đường thẳng hay mặt phẳng tuyến tính có thể giúp phân loại các điểm dữ liệu (Hình 3). Hàm ánh xạ thường không cố định và được lựa chọn tùy theo đặc tính của dữ liệu và tính chất bài toán.

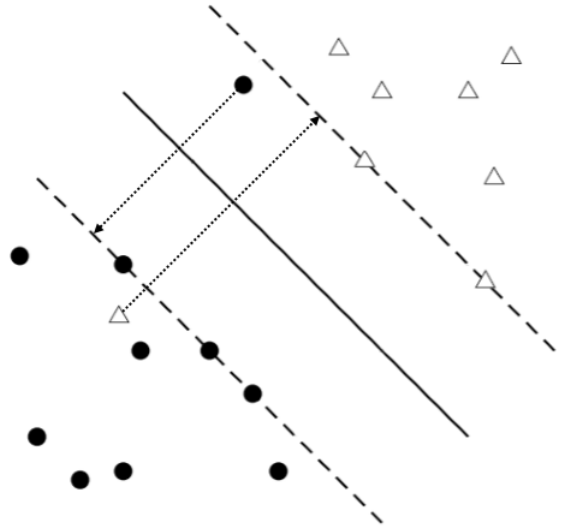


Hình 3: Minh họa ánh xạ tập dữ liệu không khả phân cách tuyến tính [1]

Phương pháp soft-margin

Trên thực tế, sai số của dữ liệu đầu vào là một trong những nguyên nhân dẫn đến bài toán phân loại phi tuyến. Trong trường hợp này, phương pháp *soft-margin* thường được sử dụng để cải thiện giải thuật tối ưu hóa khoảng cách (Hình 4). Phương pháp này cho phép tồn tại một số điểm được phân chia sai lớp với một giới hạn sai số nhất định (gọi là độ lỗi).

[1] Tham khảo Figure 7 của [26]



Hình 4: Minh họa phương pháp soft-margin

Bài toán phân loại đa lớp

Mô hình SVM dạng chuẩn như trên chỉ giúp phân loại dữ liệu thành hai lớp. Trong khi đó, bài toán thực tế đòi hỏi số lượng lớp phân loại đầu ra thường lớn hơn 2. Ví dụ như bài toán phân tích cảm xúc trên văn bản y khoa yêu cầu phân loại cảm xúc thành 3 lớp: *Tích cực*, *Tiêu cực* và *Trung tính*. Lúc này cần áp dụng một hoặc kết hợp một số phương pháp phân loại đa lớp sử dụng mô hình SVM như OvA (One-vesus-All), OvO (One-against-one), DDAG (Decision Directed Acyclic Graph).

Bảng 1: Minh họa phương pháp phân loại OvA cho bài toán phân tích cảm xúc trong bệnh án điện tử

Bộ phân loại	***Tích cực***	***Tiêu cực***	***Trung tính***
SVM_1	O	X	X
SVM_2	X	O	X
SVM_3	X	X	O

Bảng 1 minh họa phương pháp phân loại OvA: kết hợp nhiều mô hình SVM để phân loại dữ liệu. Trong đó, mỗi SVM giúp phân loại một lớp dữ liệu tương ứng với các lớp khác. Cụ thể với bài toán phân tích cảm xúc trong văn bản y khoa đã đề cập, ta cần ba mô hình SVM: SVM_1 phân loại dữ liệu thành lớp *Tích cực* với lớp *Không tích cực* (bao gồm *Tiêu cực* và *Trung tính*), SVM_2 phân loại dữ liệu thành lớp *Tiêu cực* với lớp *Không tiêu cực* (bao gồm *Tích cực* và *Trung tính*), và tương tự với SVM_3. Khi xuất hiện một điểm dữ liệu mới, dữ liệu đó sẽ được phân loại qua tất cả những lớp SVM đã được xây dựng.

3.4 Phương pháp đánh giá độ đồng nhất Cohen's kappa

Hệ số *Cohen's kappa*, gọi tắt là *kappa*, ký hiệu κ, là hệ số đánh giá mức độ đồng ý của 2 ý kiến đánh giá trên cùng 1 tập các đối tượng được đánh giá. Trong nghiên cứu này, chúng tôi sử dụng κ để đánh giá mức độ đồng ý của 2 người đánh nhãn trên tập dữ liệu. Phương pháp này được đánh giá cao bởi vì κ có xem xét đến xác xuất ngẫu nhiên xảy ra

Bảng 2: Thống kê các câu được đánh nhãn

	Người đánh nhãn 1				
Người đánh nhãn 2		**Tích cực**	**Tiêu cực**	**Trung tính**	**Tổng cộng**
	Tích cực	p_{11}	p_{12}	p_{13}	p_{1a}
	Tiêu cực	p_{21}	p_{22}	p_{23}	p_{2a}
	Trung tính	p_{31}	p_{32}	p_{33}	p_{3a}
	Tổng cộng	p_{1b}	p_{2b}	p_{3b}	p (Tổng số câu)

sự đồng ý giữa 2 ý kiến đánh giá.

Phương pháp hệ số *kappa* có 3 phiên bản:

- Phiên bản gốc do Jacob Cohen giới thiệu năm 1960, thường được gọi là *Cohen's kappa*. Trong nghiên cứu này, chúng tôi sử dụng phiên bản gốc.
- Phiên bản *kappa* có trọng số, giúp xét đến cả tỉ lệ không đồng ý.
- Phiên bản *Fleiss' kappa* có thể đo độ đồng nhất với số lượng người đánh nhãn không giới hạn, trong khi phiên bản gốc chỉ có thể đánh giá khi có đúng 2 người đánh nhãn.

Mỗi người đánh nhãn được yêu cầu phân loại mỗi câu thuộc vào 1 trong 3 loại: *Tích cực*, *Tiêu cực* hoặc *Trung tính*. Từ đó, các câu đã đánh nhãn được thống kê như Bảng 2. Hệ số kappa được tính bởi công thức:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

Trong đó:

- p_o là tỉ lệ đồng ý tương đối giữa 2 người đánh nhãn.

$$p_o = \frac{p_{11} + p_{22} + p_{33}}{p}$$

- p_e là giả thiết xác suất ngẫu nhiên 2 người đánh nhãn có chung 1 ý kiến. Xét các câu được đánh nhãn *Tích cực*, xác suất để người thứ nhất phân loại 1 câu thuộc lớp *Tích cực* là p_{1b}/p, xác suất để người thứ 2 phân loại 1 câu thuộc lớp *Tiêu cực* là p_{1a}/p, do đó, xác xuất ngẫu nhiên 2 người cùng phân loại 1 câu thuộc lớp *Tích cực* là:

$$\frac{p_{1b}}{p} * \frac{p_{1a}}{p}$$

  Từ đó, p_e là tổng xác suất ngẫu nhiên 2 người cùng đánh 1 câu thuộc nhãn *Tích cực*, *Tiêu cực*, *Trung tính*:

$$p_e = \frac{p_{1a}}{p} * \frac{p_{1b}}{p} + \frac{p_{2a}}{p} * \frac{p_{2b}}{p} + \frac{p_{3a}}{p} * \frac{p_{3b}}{p}$$

Nếu 2 người đánh nhãn hoàn toàn đồng ý với nhau, $p_o = 1$ và $p_e = 0$, suy ra $\kappa = 1$ là giá trị lớn nhất có thể có. Ngược lại, $\kappa < 0$ nếu tỉ lệ đồng ý giữa 2 người đánh nhãn thấp hơn cả xác xuất đồng ý ngẫu nhiên. Theo nghiên cứu [33], độ đồng nhất giữa 2 người đánh nhãn được đánh giá như Bảng 3.

Bảng 3: Thang đo đánh giá độ đồng nhất dựa trên giá trị κ

Giá trị	**Mức độ đồng nhất**
$\kappa < 0$	Thấp hơn xác xuất đồng ý ngẫu nhiên
$0.1 < \kappa < 0.2$	Hơi đồng ý (*slight*)
$0.21 < \kappa < 0.40$	Mức độ khá (*fair*)
$0.41 < \kappa < 0.60$	Mức độ vừa phải (*moderate*)
$0.61 < \kappa < 0.80$	Mức độ tốt (*substantial*)
$0.81 < \kappa < 0.99$	Mức độ gần như hoàn hảo (*almost perfect*)

3.5 Các thư viện và công cụ hỗ trợ

Thư viện UMLS

UMLS (*Unified Medical Language System*) là hệ thống từ vựng y sinh do Thư viện Y khoa Quốc gia Hoa Kỳ xây dựng từ năm 1986 bao gồm tập dữ liệu lớn các thông tin y khoa đã được chuẩn hóa và những công cụ hỗ trợ tương tác với hệ thống máy tính. Tính đến năm 2004, bộ từ điển UMLS tích hợp hơn 2 triệu tên gọi cho khoảng 900.000 khái niệm y sinh và khoảng 12 triệu tên gọi cho quan hệ giữa các khái niệm này [5]. Hiện nay UMLS vẫn liên tục được cập nhật và cho phép sử dụng miễn phí phục vụ mục đích nghiên cứu khoa học.

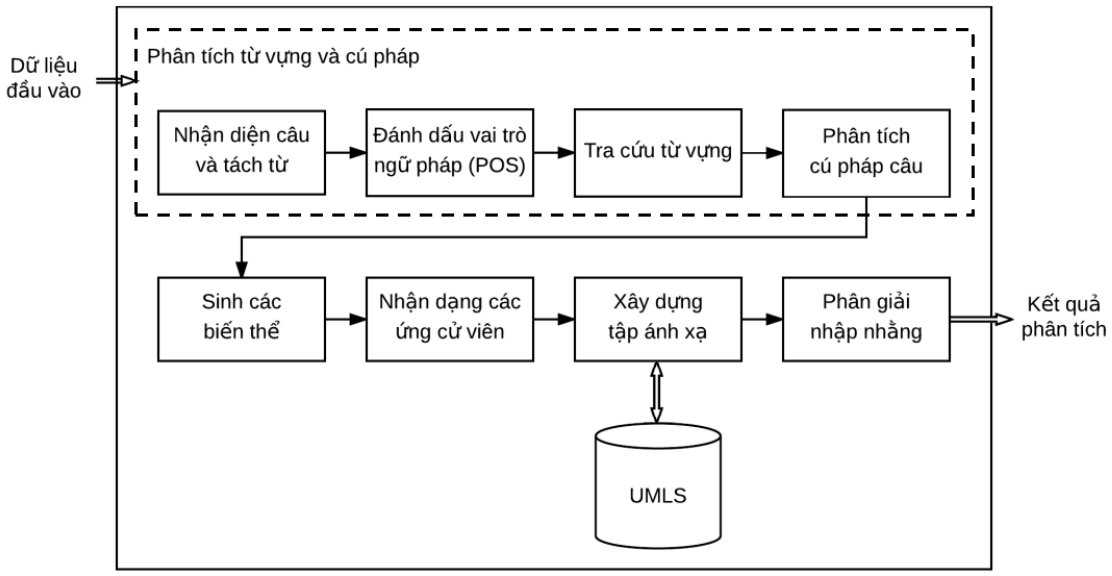
Hệ thống UMLS chứa 3 thành phần chính:

- Kho dữ liệu Metathesaurus[1]: là bộ từ điển y sinh lớn chứa các thông tin như mã số, ngữ nghĩa của các loại từ vựng, thuật ngữ y học và nhãn liên kết giữa các từ vựng khác nhau có cùng khái niệm. Metathesaurus là thành phần lớn nhất của UMLS, được tích hợp từ mạng ngữ nghĩa và các công cụ xử lý ngôn ngữ tự nhiên trong UMLS. Kho dữ liệu Metathesaurus bao gồm nhiều thư viện y khoa được sử dụng phổ biến như MeSH[2], SNOMED CT[3],...
- Mạng ngữ nghĩa (*Semantic Network*) chứa danh mục các loại ngữ nghĩa và mối quan hệ giữa chúng.
- Công cụ từ vựng (*SPECIALIST Lexicon and Lexical Tools*) bao gồm các công cụ xử lý ngôn ngữ tự nhiên được dùng để tích hợp Metathesaurus.

Công cụ MetaMap

MetaMap là công cụ hỗ trợ việc nhận dạng các khái niệm trong bộ từ điển UMLS-Metathesaurus từ văn bản y khoa. MetaMap nhận dữ liệu đầu vào là văn bản phi cấu trúc, thuần ngôn ngữ tự nhiên như các loại báo cáo y khoa, văn bản khám lâm sàng,... Sau quá trình xử lý, đầu ra của MetaMap là văn bản có cấu trúc - chủ yếu ở định dạng XML hoặc một số định dạng khác như MMO, HR - chứa các khái niệm nhận dạng được trong kho dữ liệu Metathesaurus từ văn bản đầu vào. Kiến trúc tổng quát của MetaMap được mô tả như Hình 5.

[1]https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/
[2]https://www.ncbi.nlm.nih.gov/mesh
[3]https://www.nlm.nih.gov/healthit/snomedct/



Hình 5: Kiến trúc tổng quát của MetaMap [2]

Quá trình xử lý của MetaMap có thể được tóm tắt qua 2 bước:

1. Phân tích từ vựng và cú pháp: dữ liệu đầu vào được áp dụng các tác vụ xử lý ngôn ngữ tự nhiên cơ bản như tách câu, tách từ, xác định từ loại, tra cứu từ vựng dùng công cụ từ vựng của UMLS và phân tích cú pháp câu. Sau những tác vụ này, kết quả thu được là tập hợp các cụm từ (*phrase*) được xác định từ văn bản đầu vào.

2. Phân tích chuyên sâu: ứng với mỗi cụm từ đã tìm được, MetaMap tiến hành tìm tất cả những biến thể của cụm, xác định các ứng cử viên (candidate) từ các khái niệm trong UMLS khớp với các biến thể được sinh ra và đánh giá độ tin cậy của từng ứng viên. Kết quả thu được là tập hợp các cụm đã có từ bước 1, và thông tin các ứng cử viên tương ứng.

Ví dụ với đầu vào là câu: `He denied chest pain, shortness of breath or cough.` Kết quả sau khi phân tích của MetaMap trả về như Hình 6. Các ứng cử viên được gọi tên là Meta Mapping. Câu văn được tách thành các cụm: `He`, `denied`, `chest pain`, `shortness of breath`, `or`, `cough`. Trong đó:

- Các cụm `He`, `or` không có Meta Mapping do không tìm được ứng cử viên nào.
- Cụm `denied` có 2 Meta Mapping cùng số điểm tin cậy (1000), tương ứng với 2 khái niệm trong UMLS Metathesaurus với 2 mã định danh CUI (*Concept Unique Identifier*) là C0332319 và C2700401, kèm theo là định nghĩa, mô tả và nhóm của khái niệm đó. Với cụm `denied`, MetaMap xác định được 2 nhóm là Khái niệm định tính (*Qualitative Concept*) và Hành động (*Activity*).
- Tương tự như cụm `denied`, các cụm `chest pain`, `shortness of breath`, `chest pain` cũng có 2 Meta Mapping với đầy đủ mã số CUI, định nghĩa, mô tả và nhóm của từng khái niệm.

```
Processing 00000000.tx.1: He denied chest pain, shortness of breath or cough.

Phrase: He
>>>>> Phrase
<<<<< Phrase

Phrase: denied
>>>>> Phrase
denied
<<<<< Phrase
>>>>> Mappings
Meta Mapping (1000):
  1000   C0332319:Denied (Denied (qualifier)) [Qualitative Concept]
Meta Mapping (1000):
  1000   C2700401:Denied (Deny (action)) [Activity]
<<<<< Mappings

Phrase: chest pain,
>>>>> Phrase
chest pain
<<<<< Phrase
>>>>> Mappings
Meta Mapping (1000):
  1000 N C0008031:CHEST PAIN (Chest Pain) [Sign or Symptom]
Meta Mapping (1000):
  1000   C2926613:Chest pain (Chest pain:Finding:Point in
time:^Patient:Ordinal) [Clinical Attribute]
<<<<< Mappings

Phrase: shortness of breath
>>>>> Phrase
shortness of breath
<<<<< Phrase
>>>>> Mappings
Meta Mapping (1000):
  1000 N C0013404:SHORTNESS OF BREATH (Dyspnea) [Sign or Symptom]
Meta Mapping (1000):
  1000   C2707305:Shortness of breath (Shortness of breath:-:Point in
time:^Patient:-) [Clinical Attribute]
<<<<< Mappings

Phrase: or
>>>>> Phrase
<<<<< Phrase

Phrase: cough.
>>>>> Phrase
cough
<<<<< Phrase
>>>>> Mappings
Meta Mapping (1000):
  1000 N C0010200:COUGH (Coughing) [Sign or Symptom]
Meta Mapping (1000):
  1000 N C1961131:Cough (Cough Adverse Event) [Finding]
<<<<< Mappings
```

HÌNH 6: Ví dụ kết quả chạy MetaMap

Không chỉ xác định nhóm ngữ nghĩa của các khái niệm, MetaMap còn hỗ trợ phân tích các yếu tố phủ định có trong dữ liệu đầu vào. Khi chọn bộ lọc `-negex`, kết quả phân tích sẽ thêm ký tự `N` vào trước khái niệm bị phủ định. Ví dụ như Hình 6, cụm `chest pain` có 1 Meta Mapping với mã khái niệm C0008031, thuộc nhóm Dấu hiện hoặc triệu chứng (*Sign or Symptom*) bị phủ định.

4 Phương pháp đề xuất

Trong phần này chúng tôi sẽ đặc tả chi tiết bài toán và mô hình hóa phương pháp đề xuất để xây dựng hệ thống phân tích cảm xúc trong văn bản y khoa. Đồng thời chúng tôi cũng trình bày cụ thể các thành phần trong hệ thống và cách sử dụng các thành phần này để giải quyết bài toán.

4.1 Mô tả bài toán

Nghiên cứu này giải quyết bài toán: Phân loại cảm xúc của câu trong các báo cáo nghiên cứu thuộc lĩnh vực y khoa. Mỗi câu được hệ thống phân vào 1 trong 3 loại: *Tích cực*, *Tiêu cực*, hoặc *Trung tính*. Bản thân việc xác định câu trong đoạn cũng là 1 bài toán, thường được gọi là Định hướng ranh giới câu (*Sentence boundary disambiguation*), không nằm trong phạm vi đề tài. Chúng tôi sử dụng giải thuật phân tách câu Punkt sentence segmenter [18] được hiện thực trong thư viện NLTK.

Định nghĩa 3 loại cảm xúc như sau:

- *Tích cực* là những câu thể hiện kết quả tốt hơn, cải thiện hơn hoặc kết quả tích cực vượt trội so với tổng thể dù vẫn có tác dụng phụ tiêu cực.

  Ví dụ 1

  ```
  Patients subjectively reported significantly greater
  relief from symptoms with Debacterol than with
  Kenalog-in-Orabase or no treatment.
  ```

- *Tiêu cực* là những câu thể hiện kết quả xấu, tệ hơn hoặc thể hiện phương pháp không đem lại hiệu quả.

  Ví dụ 2

  ```
  There is a suggestion that routine surgical interfer-
  ence may be harmful by increasing the risk of cae-
  sarean section, and this agrees with data from other
  trials.
  ```

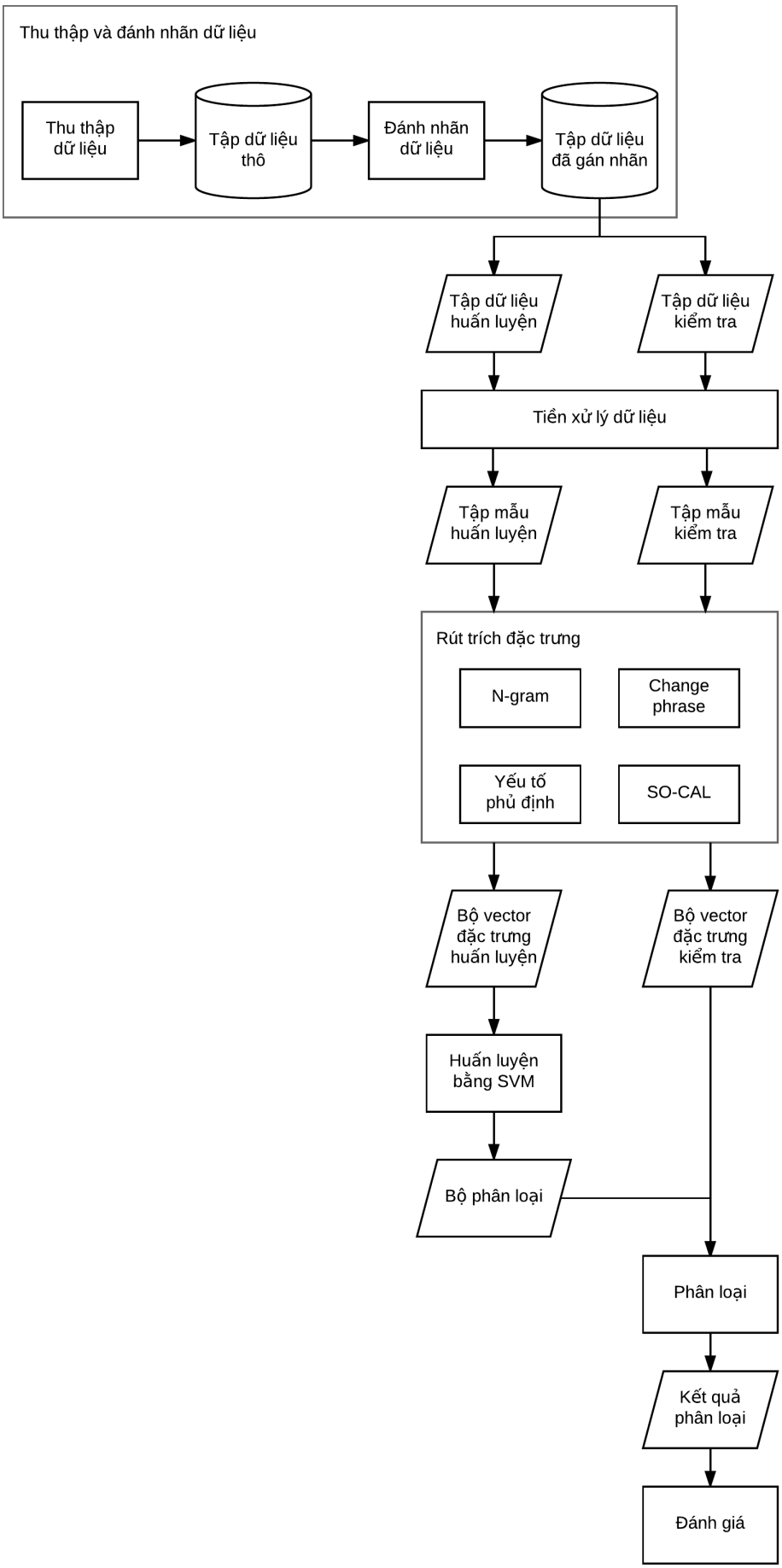
- *Trung tính* là những câu không thể hiện kết quả, không có khẳng định tốt hay xấu; hoặc đồng thời nhiều ý kiến tốt xấu mà không có sự lất át rõ ràng.

  Ví dụ 3

  ```
  Data extraction and analyses and quality assessment
  were conducted according to the Cochrane standards.
  ```

4.2 Kiến trúc tổng quan

Chúng tôi đề xuất xây dựng hệ thống phân tích cảm xúc trong báo cáo y khoa theo kiến trúc được mô tả ở Hình 7. Hệ thống gồm 5 thành phần (*modules*) chính:



HÌNH 7: Kiến trúc tổng quan xây dựng hệ thống

Thu thập và đánh nhãn dữ liệu: Nhóm tiến hành thu thập dữ liệu từ nguồn web, lưu vào hệ cơ sở dữ liệu, sau đó tiến hành đánh nhãn. Chi tiết được trình bày tại Mục 6.1.

Tiền xử lý dữ liệu: Dữ liệu thu thập được có thể gặp lỗi và có định dạng không đúng, cần được xử lý trước khi trích xuất đặc trưng.

Đây là bước xử lý trước khi có thể rút trích đặc trưng. Trong bước này, chúng tôi xử lý dữ liệu theo thứ tự:

- Chuyển tất cả các ký tự thành chữ thường.
- Xóa các ký tự đặc biệt, gồm: ?, %, @, #, ∧, $, ., „, ;, :, /, ", (,), +, -, =
- Thay tất cả số bằng nhãn *DIGIT*.
- Loại bỏ *Stop words*: Các từ *stop word* là những từ thông thường được sử dụng mà có tính chất phân cực thấp. Một số từ *stop word* như: it, I, you, then, ...
- *Tokenization*: Sử dụng ký tự khoảng trắng để tách tách câu thành các token
- *Lemmatization:* Trả về dạng đúng của một động từ, dù động từ đó đang ở thì nào. *Lemmatization* thực hiện bằng cách loại bỏ đi các biến tố (*inflectional*).
  Ví dụ: “produced” -> “produce”.
- *Stemming* Loại bỏ hầu hết các hậu tố của 1 từ, trả về gốc của một từ dù từ đó được dùng như động từ, tính từ, danh từ hay phó từ.
  Ví dụ: “produced” hoặc “production” -> “produc”.

Rút trích đặc trưng: Có 4 đặc trưng được sử dụng: N-gram, change phrase, sự phủ định và SO-CAL được trình bày trong các Mục từ 4.3 đến 4.6.

Huấn luyện: Thực hiện việc huấn luyện với giải thuật học máy SVM để tạo ra bộ phân loại.

Phân loại: Sử dụng bộ phân loại từ khối Huấn luyện, với mỗi dữ liệu đầu vào, bộ phân loại sẽ phân loại câu thuộc 1 trong 3 lớp: *Tích cực*, *Tiêu cực*, hoặc *Trung tính*

Đánh giá: Sử dụng khối phân loại ở trên, áp dụng đầu vào là tập dữ liệu kiểm tra để đánh giá hiệu quả của bộ phân loại.

Trong phần còn lại, chúng tôi trình bày các đặc trưng theo cấu trúc 2 phần: Phần Mô tả trình bày ý tưởng, mô tả về đặc trưng, phần Rút trích trình bày cách sử dụng đặc trưng để có thể áp dụng vào hệ thống.

4.3 N-gram

Mô tả

Theo kết luận của nghiên cứu [6], n-gram là đặc trưng được sử dụng phổ biến trong bài toán phân tích cảm xúc nói chung. Nhiều nghiên cứu về phân tích cảm xúc trong lĩnh vực y khoa cũng sử dụng đặc trưng này [24], [22], [28], [21], [27], [34]

N-gram là một chuỗi gồm n phần tử liên tiếp nhau. Các phần tử này có thể chữ cái, âm tiết hoặc đoạn văn... Trong nghiên cứu này, các phần tử là các từ đơn trong câu. Đơn vị

từ được định nghĩa là chuỗi các chữ cái liên tiếp nhau không chứa ký tự khoảng trắng, các từ phân biệt nhau bởi ký tự khoảng trắng. Đặc trưng n-gram đóng góp lớn trong kết quả của các phân tích, vì vậy đặc trưng này thường được dùng như baseline. Báo cáo của [24] phân tích cảm xúc trên các bình luận về phim, đạt độ chính xác 82.9% với chỉ một đặc trưng n-gram. Đây cũng là kết quả tốt nhất của nghiên cứu này. Nghiên cứu của [22] đạt độ chính xác 77.87% khi chỉ sử dụng n-gram như *baseline*, kết quả tốt nhất tăng 20.58% so với *baseline*.

Ví dụ câu: `Standard practice in pupillary monitoring yields inaccurate data`

- Với $n = 1$, n-gram được gọi là uni-gram. Câu trên sẽ được chuyển thành các n-gram: Standard, practice, in, pupillary, monitoring, yields, inaccurate, data.
- Với $n = 2$, n-gram được gọi là bi-gram. Câu trên sẽ được chuyển thành các n-gram: Standard practice, practice in, in pupillary, pupillary monitoring, monitoring yields, yields inaccurate, inaccurate data.
- Với $n = 3$, n-gram được gọi là tri-gram. Câu trên sẽ được chuyển thành các n-gram: Standard practice in, practice in pupillary, in pupillary monitoring, pupillary monitoring yields, monitoring yields inaccurate, yields inaccurate data.
- Với $n > 3$, tần suất xuất hiện các n-gram thấp, dễ làm mô hình học máy bị học quá khớp (overfitting)

Việc phối hợp các n-gram là tùy chọn đối với mỗi nghiên cứu, và các kết quả cũng không hoàn toàn đồng nhất. Báo cáo [22] kết luận khi sử dụng bi-gram kết hợp với uni-gram giúp tăng độ chính xác thêm 3.01%, điều này phù hợp với báo cáo [28]. Báo cáo [28] kết luận rằng việc dùng cả uni-gram, bi-gram và tri-gram giúp cải thiện kết quả rõ rệt. Trong khi đó [24] đạt kết quả cao nhất chỉ với uni-gram. Kết luận của [24] cho thấy việc sử dụng thêm đặc trưng bi-gram không tác động nhiều đến kết quả. Báo cáo của [30] phân tích cảm xúc trong lĩnh vực y khoa lâm sàng, khẳng định đặc trưng uni-gram cho kết quả tốt hơn bi-gram. Tuy nhiên, nghiên cứu không thử nghiệm kết hợp 2 đặc trưng này. Trong báo cáo này, chúng tôi thử nghiệm các cách kết hợp khác nhau của n-gram để tìm ra kết quả tốt nhất.

TODO Việc áp dụng các kiến thức liên quan đến lĩnh vực đang xem xét giúp tăng độ chính xác khi phân loại. Tác giả Kerstin Denecke trong nghiên cứu [12] khẳng định rằng áp dụng kiến thức trong lĩnh vực y khoa là cần thiết để cải thiện hiệu quả phân loại cảm xúc. Trong nghiên cứu này, ý nghĩa cụ thể của các thuật ngữ y khoa không có tác dụng phân loại cảm xúc, chỉ thông tin mô tả của các thuật ngữ này có ý nghĩa.

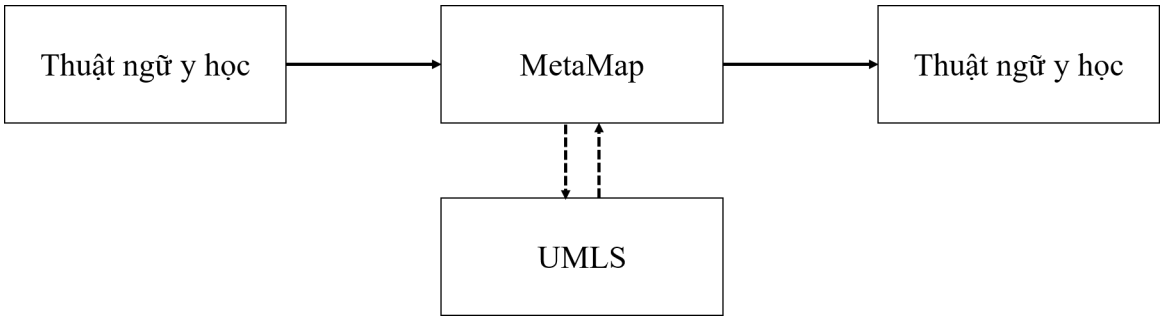
Ví dụ

```
Elevated troponin level after acute stroke is common
and is associated with ECG changes suggestive of my-
ocardial ischemia and increased risk of death
```

Từ `stroke` xét trong ngữ cảnh y học nghĩa là đột quỵ. Nhưng đối với việc phân loại cảm xúc, chúng tôi chỉ quan tâm tới ý nghĩa khái quát của từ này: `stroke` mô tả một loại bệnh. Tương tự các từ `diarrhoea, abdominal pain, nausea` chỉ cần được hiểu như vấn đề về bụng mà không cần hiểu cụ thể như thế nào. Như vậy, các thuật ngữ y khoa thuộc cùng 1 kiểu (triệu chứng, loại bệnh, tên thuốc, ...) đều được xem là một. Từ

đó, giảm thiểu khả năng bộ phân loại bị nhiễu hoặc bị học lệch (*overfitting*).

Một trong những công cụ đã được xây dựng hoàn chỉnh và sử dụng phổ biến trong các bài toán liên quan đến y khoa trên dữ liệu tiếng Anh là UMLS. Đây là một hệ thống tích hợp các thuật ngữ y khoa cùng các mã chuẩn hóa nhằm tạo tiền đề cho việc xây dựng và phát triển các hệ thống thông tin y khoa cũng như các dịch vụ chăm sóc y tế khác. Để hiện thực nhiệm vụ trên, chúng tôi sử dụng công cụ MetaMap kết hợp hệ thống UMLS. UMLS phân các thuật ngữ y học ra làm 136 kiểu [1]. MetaMap là một công cụ cho phép tra cứu tên kiểu của 1 thuật ngữ y học bất kỳ (Hình 8).



HÌNH 8: MetaMap sử dụng nguồn tài nguyên UMLS, giúp tra cứu tên kiểu của 1 thuật ngữ y học

Ví dụ 1

`Elevated troponin level after acute stroke is common and is associated with ECG changes suggestive of myocardial ischemia and increased risk of death`

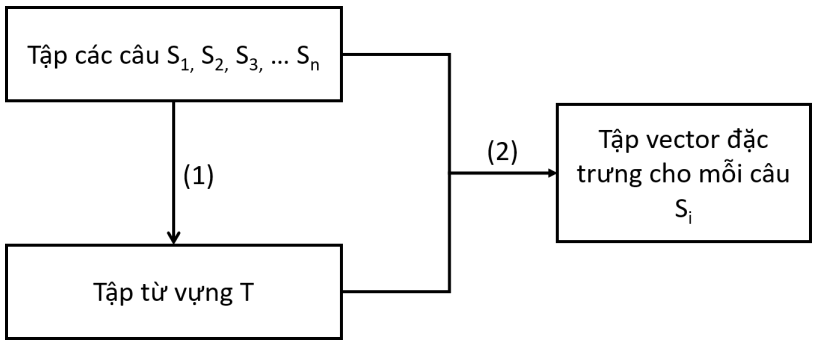
$\xrightarrow{MetaMap}$

`Elevated troponin level after acute DSYN is common and is associated with ECG changes suggestive of myocardial DSYN and increased risk of death`

Từ `stroke` và `ischemia` đều thuộc kiểu loại bệnh hoặc triệu chứng, nên được thay bằng nhãn DSYN (Disease or Syndrome)

Rút trích

Giải thuật trích xuất đặc trưng n-gram trải qua 2 bước, được mô tả như Hình 9



HÌNH 9: Giải thuật trích xuất đặc trưng n-gram

[1] Chi tiết các kiểu tham khảo tại `https://metamap.nlm.nih.gov/Docs/SemanticTypes_2013AA.txt`

Ở bước đầu tiên (1), giải thuật nhận vào tập hợp các câu. Mỗi câu sẽ được tách ra thành các n-gram. Tất cả các n-gram từ các câu sẽ được tổng hợp lại thành tập từ vựng T. Tuy nhiên, không phải tất cả các n-gram đều được thêm vào tập từ vựng T. Giải thuật quy định 1 mức ngưỡng *min_df* là số câu tối thiểu cùng chứa 1 n-gram thì n-gram đó mới được thêm vào tập T. Nếu $min\_df = 1$ thì tập từ vựng T chứa tất cả các n-gram.

Nghiên cứu [28] sử dụng $min\_df = 5$, trong khi nghiên cứu [22] dùng $min\_df = 4$. Tuy nhiên cả 2 nghiên cứu trên đều không giải thích về cách chọn các giá trị trên. Trong nghiên cứu này, chúng tôi tiến hành các thí nghiệm để phân tích và chọn ra giá trị *min_df* tối ưu nhất.

Bước còn lại (2) là *vector* hóa câu: ánh xạ 1 câu từ dạng *text* sang dạng *vector* đại diện cho câu đó. Các n-gram trong tập từ vựng T ở bước (1) được tiến hành sắp xếp. Việc sắp xếp này là tùy ý, nhưng sau khi đã sắp xếp phải giữ nguyên thứ tự. Giả sử $T = \{\text{n-gram}_1, \text{n-gram}_2, \text{n-gram}_3, \ldots, \text{n-gram}_n\}$. Khi đó, mỗi câu s_i sẽ được chuyển thành *vector* v_i có n chiều. Có 2 cách hiện thực để xác định giá trị tại chiều thứ k của *vector* v_i:

- Giá trị tại chiều thứ k của vector v_i là giá trị nhị phân, bằng 0 nếu câu đó không chứa n-gram_k, bằng 1 nếu câu đó chứa n-gram_k.
- Giá trị tại chiều thứ k của vector v_i là số nguyên, thể hiện số lần xuất hiện n-gram_k trong câu đó.

Để thuận tiện khi gọi tên trong các thử nghiệm, chúng tôi quy ước đặt tên cách hiện thực thứ nhất là *vector nhị phân*, cách hiện thực thứ 2 là *vector số nguyên*.

Ví dụ 3

> Giả sử sau khi qua bước (1), thu được tập từ vựng T gồm các n-gram: drug, risk, disturbances, associated with, disadvantage, evidence. Khi đó, nếu sử dụng cách vector hóa dùng *vector nhị phân*, các câu ở ví dụ 1 và 2 được chuyển thành dạng vector như sau:

	drug	**risk**	**disturbances**	**associated with**	**disadvantage**	**evidence**
Ví dụ 1	0	1	0	1	0	0
Ví dụ 2	0	1	0	0	1	0

Khi đó, $v_1 = (0, 1, 0, 1, 0, 0)$ và $v_2 = (0, 1, 0, 0, 1, 0)$

4.4 Change Phrase

Mô tả

Đặc trưng change phrase được Niu, Yun et al. định nghĩa trong một nghiên cứu phân tích cảm xúc trên câu [22]. Sau đó được nhóm tác giả Sarker, Abeed, et al. sử dụng lại. Bài toán mà Sarker, Abeed, et al giải quyết cũng tương tự nhưng thay vì phân tích trên câu, nhóm tác giả phân tích trên đoạn. Ngoài việc sử dụng lại, Saker, Abeed,et al. có một số thay đổi và mở rộng đặc trưng này.

Change phrase là những cụm từ mang ý nghĩa làm thay đổi tình trạng, trạng thái: làm tốt hơn hoặc làm tệ hơn. Tính phân cực trong một câu thường biểu thị qua sự thay đổi [22], và hay xuất hiện ở những câu so sánh.
Ví dụ

```
Atypical antipsychotic use is associated with an in-
creased risk for death compared with nonuse among
older adults with dementia
```

Câu trên thể hiện tình trạng tệ hơn: Sử dụng `Atypical antipsychotic` làm tăng nguy cơ chết so với không sử dụng `Atypical antipsychoti`. Chúng tôi sử dụng 4 nhóm để mô tả *Change phrase*:

- Nhóm thể hiện sự thay đổi tình trạng, gồm 2 nhóm:
  LESS: Có ý nghĩa làm giảm bớt, hạ bớt. Một số từ như: "reduce", "decline", "fall", "less", "little",...
  MORE: Có ý nghĩa ngược lại, làm tăng thêm (hoặc duy trì). Một số từ như: "enhance", "higher", "exceed", "increase", "improve",...

- Nhóm xác định tính phân cực, gồm 2 nhóm:
  GOOD: Mang ý nghĩa tích cực. Một số ví dụ như: "benefit", "improvement", "advantage", "accuracy", "great", ...
  BAD: Mang ý nghĩa tiêu cực. Một số ví dụ như: "suffer", "adverse", "hazards", "risk", "death",...

Danh sách các từ cho mỗi nhóm trên được chúng tôi tập hợp thủ công. Kết hợp 4 nhóm trên, ta có 4 đặc trưng giúp mô tả những thay đổi tích cực hoặc tiêu cực như Bảng 4.
Ví dụ

```
Atypical antipsychotic use is associated with an in-
creased risk for death compared with nonuse among
older adults with dementia
```

Từ `increased` sẽ được gán nhãn MORE, `risk` được gán nhãn BAD, sau đó việc phân tích sẽ xác định được đối tượng của từ `increased` là `risk`. Từ đó, câu trên được nhận dạng thuộc mẫu MORE-BAD, suy ra nó có xu hướng biểu thị tính phân cực *Tiêu cực*.

Bảng 4: Các đặc trưng *Change phrase*

Nhóm làm thay đổi tình trạng	**Nhóm xác định đối tượng**	**Phân loại tính phân cực**
LESS	GOOD	*Tiêu cực*
LESS	BAD	*Tích cực*
MORE	GOOD	*Tích cực*
MORE	BAD	*Tiêu cực*

Rút trích

Rút trích đặc trưng Change phrase phụ thuộc vào 2 yếu tố:

- Danh sách từ trong mỗi nhóm LESS, MORE, BAD, GOOD
- Giải thuật nhận biết sự kết hợp của các nhãn trên

Với yếu tố thứ nhất, trong nghiên cứu này, chúng tôi sử dụng danh sách từ cho mỗi nhóm tham khảo từ nghiên cứu của nhóm tác giả Sarker, Abeed, et al.[28]. Nhóm tác giả trên tự tập hợp danh sách các nhóm từ thủ công nhưng không liệt kê trong báo cáo của mình. Nhóm chúng tôi có liên hệ và đã nhận được mã nguồn, từ đó lấy được danh sách các nhóm từ. Danh sách này gồm 371 từ (BAD: 223 từ, GOOD: 82 từ, MORE: 30 từ, LESS:36 từ).

Sau đó, chúng tôi mở rộng danh sách bằng cách thu thập thủ công. Danh sách cuối cùng gồm 423 từ (BAD: 238, GOOD: 96, MORE: 42 từ, LESS: 47 từ).

Sau khi đã có tập hợp các từ cho mỗi nhóm, chúng tôi xem xét yếu tố thứ 2: hiện thực giải thuật nhận dạng xem 1 câu có thuộc mẫu nào trong 4 mẫu: LESS-GOOD, LESS-BAD, MORE-GOOD, MORE-BAD. Giải thuật nhận dạng câu thuộc mẫu nào được thực hiện qua 2 bước.

Ở bước 1, giải thuật nhận dạng những từ mô tả sự thay đổi, bằng cách so trùng các từ trong 2 nhóm LESS và MORE với các từ trong câu. Để có thể so trùng thành công, trước tiên các từ trong 2 nhóm này được xử lý *lemmatization* và *stemming* như ở mục **??**. Sau đó mỗi từ trong câu được so sánh với các từ trong 2 nhóm trên. Nếu từ w thuộc 1 trong 2 nhóm trên, giải thuật thêm *tag* "_LESS" hoặc "_MORE" vào cuối các từ thuộc phạm vi từ từ w đến dấu chấm câu (*punctuation*) gần nhất (về phía cuối câu). Dấu chấm câu (*punctuation*) có thể là dấu chấm (.), dấu phẩy (,), dấu hai chấm (:) hoặc dấu chấm phẩy (;). Ở bước này, đặc trưng Change phrase không thực sự tạo ra một đặc trưng mới, nó chỉ làm thay đổi các n-gram: từ risk thành risk_MORE, từ đó thay đổi tập từ vựng T của đặc trưng n-gram . Bằng cách này, thông qua đặc trưng n-gram, Change phrase không chỉ nhận biết được trong câu có sự mô tả về thay đổi, mà còn biết được phạm vi ảnh hưởng của sự thay đổi đó.

Bước 2 nhận diện xem câu có thuộc mẫu nào trong 4 mẫu: LESS-GOOD, LESS-BAD, MORE-GOOD, MORE-BAD hay không. Nếu trong câu có 1 từ thuộc nhóm MORE, giải thuật sẽ xác định trong phạm vi từ từ đó đến hết câu, nếu có từ nào thuộc nhóm GOOD, câu đó thuộc mẫu MORE-GOOD. Tương tự như vậy đối với 3 mẫu còn lại.

Cuối cùng, mỗi câu được chuyển thành 1 vector 4 chiều. Giá trị tại chiều thứ i bằng 1 nếu câu thuộc mẫu thứ i, ngược lại bằng 0. Thứ tự các mẫu được sắp xếp như sau: MORE-GOOD, MORE-BAD, LESS-GOOD, LESS-BAD

4.5 Thành phần phủ định

Mô tả

Bài toán phân tích phủ định bao gồm hai nhiệm vụ chính là (1) xác định yếu tố phủ định cùng với phạm vi phủ định trong câu và (2) phân tích ảnh hưởng cũng như hiệu quả của yếu tố phủ định lên ý nghĩa phân loại tính phân cực của cả câu. Để giải quyết bài toán này chúng tôi đã tìm hiểu và hiện thực lại giải thuật phân tích phủ định NegEx[32] (chi tiết hiện thực được mô tả cụ thể ở chương 5).

Giải thuật NegEx dùng để xác định sự tồn tại của phủ định trong câu và xác định xem một cụm từ bất kỳ trong câu có chịu ảnh hưởng của yếu tố phủ định hay không. Giải thuật nhận dữ liệu đầu vào là câu văn được nghi ngờ có sự phủ định và một cụm từ thuộc câu văn đó mà cần xác định xem có bị phủ định hay không. Sau quá trình xử lý, giải thuật đưa ra câu trả lời gồm: câu văn có tồn tại sự phủ định không, xác định từ phủ định trong câu và cụm từ được hỏi có bị phủ định hay không.

Trong quá trình xử lý, NegEx dùng danh sách thuật ngữ phủ định và danh sách thuật ngữ kết thúc để giải quyết bài toán. Bên cạnh đó, giải thuật xây dựng hai biểu thức chính quy `RE` (*regular expressions*) để xác định phạm vi phủ định trong câu. Biểu thức `RE1`

bao gồm tất cả các từ (từ đơn hoặc cụm từ) đứng sau thuật ngữ phủ định và sẽ kết thúc bởi một thuật ngữ kết thúc hoặc dấu kết thúc câu hoặc một thuật ngữ phủ định khác. Biểu thức `RE2` chỉ xác định khoảng 5 từ (từ đơn hoặc cụm từ), ưu tiên lĩnh vực y khoa đứng trước thuật ngữ phủ định đang xét.

Áp dụng vào bài toán, với mỗi câu trong dữ liệu đầu vào, giải thuật NegEx lặp lại theo các bước sau:

1. Xác định tất cả các từ phủ định có trong câu dựa trên danh sách thuật ngữ phủ định, ký hiệu là tập A.
2. Tìm từ phủ định đầu tiên trong câu, kí hiệu là *Neg1*.
3. Nếu *Neg1* là từ phủ định giả, bỏ qua và thực hiện bước 6.
4. Nếu *Neg1* là từ phủ định tiền điều kiện: dùng biểu thức chính quy `RE1` xác định vùng phủ định của *Neg1*.
5. Nếu *Neg1* là từ phủ định tiền điều kiện: dùng biểu thức chính quy `RE2` xác định vùng phủ định của *Neg1*.
6. Tìm từ phủ định kế tiếp (cho đến khi hết các từ trong tập A), gán cho *Neg1* và lặp lại bước 3.

Một số ví dụ khi chạy giải thuật NegEx:
Ví dụ 8

> `Patients subjectively reported significantly greater relief from symptoms with Debacterol than with Kenalog-in-Orabase or ([no] treatment).`
> Từ phủ định tìm được: `no` là thuật ngữ phủ định tiền điều kiện, phạm vi phủ định được xác định, từ bị phủ định là `treatment`.

Ví dụ 9

> `The patient is (tumor [free]).`
> Từ phủ định tìm được: `free` là thuật ngữ phủ định hậu điều kiện, phạm vi phủ định được xác định, từ bị phủ định là `tumor`.

Rút trích

Sau khi đã xác định được từ phủ định và từ chịu ảnh hưởng phủ định trong câu, chúng tôi thực hiện rút trích đặc trưng phủ định theo 3 cách sau để áp dụng vào hệ thống:

Cách thứ nhất tham khảo từ nghiên cứu [22]. Trong nghiên cứu đó, nhóm tác giả TODO <thêm tên> chỉ xem xét 1 từ phủ định "no". Đây có thể là sự thiếu sót của nhóm tác giả, vì kết quả trong báo cáo [22] cho thấy yếu tố phủ định được thêm vào hầu như không giúp cải thiện độ chính xác. Tiếp theo, nhóm tác giả trên sử dụng công cụ Apple Pie parser để trích xuất các cụm từ, cụm từ nào có chứa từ "no" sẽ được gắn thêm hậu tố "_NO". Với cách này, yếu tố phủ định không thực sự là 1 đặc trưng mà chỉ ảnh hưởng đến hệ thống thông qua đặc trưng n-gram. Chúng tôi thử nghiệm cách này nhưng thay vì chỉ quan tâm đến từ "no", chúng tôi xem xét đến tất cả các từ được xem là từ phủ định theo giải thuật được mô tả ở phần trước. Sau đó, các từ được xem là chịu ảnh hưởng phủ định được thêm hậu tố "_NEG".
Ví dụ

Câu: `Patients subjectively reported significantly greater relief from symptoms with Debacterol than with Kenalog-in-Orabase or ([no] treatment)`
sẽ được chuyển thành:
`Patients subjectively reported significantly greater relief from symptoms with Debacterol than with Kenalog-in-Orabase or no treatment_NEG`

Cách thứ 2 chúng tôi thử nghiệm có tác dụng tương tự cách trên: không thực sự là 1 đặc trưng mà chỉ ảnh hưởng đến đặc trưng n-gram. Nhưng thay vì làm thay đổi từ bị ảnh hưởng phủ định, cách hiện thực này thay đổi từ phủ định: thay tất cả các từ phủ định bằng nhãn đại diện "NEGATION"

Ví dụ

Câu: `Patients subjectively reported significantly greater relief from symptoms with Debacterol than with Kenalog-in-Orabase or ([no] treatment)`
sẽ được chuyển thành:
`Patients subjectively reported significantly greater relief from symptoms with Debacterol than with Kenalog-in-Orabase or NEGATION treatment`

Cách còn lại cũng chỉ quan tâm đến từ phủ định, nhưng khác 2 cách trên: cách hiện thực này tạo ra 1 vector đại diện chứ không làm ảnh hưởng đến đặc trưng n-gram. Mỗi câu sẽ được ánh xạ thành 1 số nhị phân: 0 nếu câu đó không chứa yếu tố phủ định nào, 1 nếu ngược lại.

Ngoài ra, chúng tôi cũng tiến hành thử nghiệm kết hợp các cách hiện thực trên để tìm ra cách rút trích hiệu quả nhất.

4.6 Đặc trưng mở rộng SO-CAL

Mô tả

Trong nghiên cứu này, chúng tôi sử dụng đặc trưng SO-CAL dựa trên bài báo [31], tên của đặc trưng xuất phát từ tên hệ thống mà bài báo trên đã xây dựng. Theo ý nhóm tác giả [31], SO-CAL là chữ viết tắt của Semantic Orientation CALculator. Đây là một phương pháp phân tích cảm xúc dựa trên từ vựng.

Phân tích cảm xúc dựa trên từ vựng là một phương pháp khác, tránh được bất lợi của phương pháp học máy là không cần qua quá trình huấn luyện. Tuy nhiên đa số các nghiên cứu này đều phân tích cảm xúc trên văn bản thông thường, không tập trung vào 1 lĩnh vực cụ thể nào [31, 36, 23, 14].

Phương pháp này ngầm định 2 giả thiết sau đã được thỏa mãn:

- Bản thân mỗi từ có sẵn tính phân cực mà không bị phụ thuộc vào ngữ cảnh. Điều này có nghĩa là mỗi từ luôn chỉ có 1 xu hướng phân cực (tốt, xấu hoặc tích cực, tiêu cực) trong mọi câu mà từ đó xuất hiện.
- Tính phân cực của mỗi từ được đề cập ở trên có thể được biểu diễn bởi 1 số thực

Dựa trên 2 giả thiết trên, tính phân cực của một câu phụ thuộc vào số thực biểu diễn tính phân cực của các từ trong câu đó, và cũng được biểu diễn bởi 1 số thực. Sự phụ thuộc giữa tính phân cực của các từ và tính phân cực của cả câu là tùy thuộc vào các nghiên cứu, có thể mô hình hóa như công thức sau:

$$Polarity_{sentence} = f(Polarity_{words-in-sentence}) \quad (1)$$

Rút trích

Rút trích đặc trưng SO-CAl chính là giải thuật tính điểm số cho mỗi câu phụ thuộc vào điểm số của mỗi từ trong câu. Sau đây, chúng tôi trình bày các vấn đề chính khi tính điểm cho từ

Từ điển Đây là công cụ giúp tra cứu điểm số của mỗi từ. Các từ điển hiện nay thuộc 2 loại: xây dựng thủ công hoặc xây dựng bán tự động. Từ điển xây dựng thủ công điển hình được sử dụng nhiều trong các nghiên cứu là bộ từ điển General Inquirer. Lợi thế của loại tự điển này là được con người đánh điểm số, vì vậy giảm thiếu sai sót. Các nghiên cứu có thể dựa trên bộ từ điển này để tự mở rộng thành bộ từ điển của mình. Tuy nhiên, bất lợi của loại này là kích thước thường nhỏ. Bộ từ điển thuộc loại thứ 2 được sử dụng phổ biến là WordNet. Từ điển thuộc loại bán tự động được xây dựng bằng cách sử dụng một tập từ hạt giống có tính phân cực cao. Các từ này có thể được tập hợp thủ công hoặc được lấy từ từ điển General Inquirer. Từ đó, điểm của các từ khác được sinh ra dựa trên tần số xuất hiện của chúng so với các từ hạt giống. Lợi thế của loại từ điển này là kích thước lớn, tạo độ bao phủ cao, từ đó nhiều từ được gán điểm số hơn. Tuy nhiên độ chính xác của loại từ điển này không cao, và kích thước lớn thường đi kèm với nhiễu.

Nghiên cứu [31] vì thế chọn phương án xây dựng một bộ từ điển thủ công dựa trên một số nguồn text như các bình luận về phim, sách, máy tính, khách sạn,... và các từ thuộc nhóm tích cực, tiêu cực từ từ điển General Inquirer. Điểm số mỗi từ trong từ điển được xây dựng có giá trị từ -5 đến 5 với ý nghĩa: Giá trị càng nhỏ thể hiện tính phân cực về phía tiêu cực càng nhiều, và ngược lại.

Từ	**Giá trị**
monstrosity	-5
hate (noun and verb)	-4
disgust	-3
sham	-3
fabricate	-2
delay (noun and verb)	-1
determination	1
determination	1
inspire	2
inspiration	2
endear	3
relish (verb)	4
masterpiece	5

Từ loại Hầu hết các nghiên cứu ban đầu về phân tích cảm xúc dựa trên từ vừng đều chỉ tập trung vào tính từ. Điểm số của cả văn bản chỉ phụ thuộc vào điểm số của tính từ

trong câu và những từ liên quan. Xét 2 ví dụ sau:

- `The young man strolled+ purposefully+ through his neighborhood+`
- `The teenaged male strutted- cockily- through his turf-`

Các dấu + ở câu thứ nhất thể hiện xu hướng bổ sung tích tích cực vào độ phân cực của cả câu, trong khi dấu - ở ví dụ thứ 2 ngược lại. Từ đó cho thấy, ngoài tính từ, các từ loại động từ, danh từ và phó từ cũng có tác động đến điểm số phân cực của cả câu. Các nghiên cứu sau này đã mở rộng hơn, ngoài tính từ, các từ loại động từ, danh từ và phó từ cũng được xem xét tới.

Điều này không những giúp đánh giá tốt hơn trong trường hợp ví dụ trên, ngoài ra còn giúp giải quyết một số trường hợp đặc biệt khi một từ có thể thuộc nhiều loại từ loại. Ví dụ: từ `novel` nếu xét theo từ loại danh từ thì chỉ mang ý nghĩa trung tính, nếu xét theo từ loại tính từ lại có ý nghĩa tích cực. Từ `plot` mang ý nghĩa tiêu cực nếu là động từ, nhưng mang tính nghĩa trung tính nếu xem là danh từ. Trong nghiên cứu [31], bộ từ điển được xây dựng gồm 2252 tính từ, 1142 danh từ, 903 động từ, and 745 phó từ.

Tất cả các động từ và danh từ trong từ điển đều được xử lý *lemmatized*, vì vậy các động từ ở các dạng thể hiện khác nhau đều cùng 1 điểm số. Các động từ, danh từ và tính từ đều được đánh điểm thủ công, riêng phó từ được đánh điểm tự động dựa trên tính từ. Phó từ được bỏ đuôi "-ly", sau đó so trùng với các tính từ. Ví dụ: từ `purposefully` sẽ được gán điểm số của tính từ `purposefull`. Tuy nhiên, một số ngoại lệ như các phó từ `fast` được xử lý thủ công.

Tính tăng cường (intensification) Một từ có tính tăng cường được hiểu là các từ bản thân nó không có điểm số thể hiện tính phân cực, nhưng có khả năng tác động lên 1 từ khác, làm tăng lên hoặc hạ thấp tính phân cực của từ đó. Từ đó làm thay đổi điểm số thể hiện tính phân cực của cả cụm từ. Một số từ có tính tăng cường như: `slightly`, `very`, `most`, `the most`. Một giải thuật đơn giản có thể được sử dụng trong trường hợp này như sau: Khi gặp một từ có tác động làm tăng tính phân cực (`very`), điểm của từ bị tác động được cộng thêm 1 hằng số. Cụm từ `very good` được tính điểm bằng công thức: $Polarity(\texttt{very good}) = P(\texttt{good}) + 1$. Tương tự với trường hợp còn lại.

Tuy nhiên cách hiện thực này không thể hiện đúng bản chất của tác động tăng cường. Bởi vì cùng một từ `very` có thể có tác động mạnh yếu khác nhau tùy thuộc vào từ bị tác động. Nói cách khác, sự tác động này nên phụ thuộc vào cả 2 thành phần:

- Tính chất tăng cường của từ tác động. Tính chất này có thể được thể hiện bằng tỉ lệ phần trăm (%) như Bảng 5.
- Tính phân cực của từ bị tác động

Nghiên cứu [31] sử dụng công thức (2) để tính điểm số trong trường hợp này:

$$Polarity(\text{cụm từ}) = Polarity(\text{từ bị tác động}) * (100\% - \text{tỉ lệ tác động}) \quad (2)$$

Bảng 5: Tỉ lệ tác động của một số từ

Từ	**Tỉ lệ tác động**
slightly	-50%
somewhat	-30%
pretty	-10%
really	+15%
very	+25%
extraordinarily	+50%
(the) most	+100%

Ví dụ

`good` có điểm số là 3.0, từ đó `very good` có điểm số là: 3.0 * (100% + 25%) = 3.75

Trong trường hợp có hơn 1 từ có tính tăng cường, điểm số được tính tương tự theo cách đệ quy.

Ví dụ

`really very good`: (3 * [100% + 25%]) * (100% + 15%) = 4.3

Trong trường hợp một tính từ bổ sung nghĩa cho một danh từ theo sau nó, tính từ đó được coi như là một từ có tính tăng cường. Vì vậy, bản thân tính từ đó không có điểm số, và chỉ làm thay đổi điểm số của danh từ theo sau.

Ví dụ

`This is a total failure`
Tính từ `total` được xem là từ có tính tăng cường, nên thay vì sử dụng điểm số, `total` ảnh hưởng đến điểm số của `failure`: $-3.0*(100\%+50\%) = -4.5$

Xử lý phủ định Sự xuất hiện từ phủ định có thể làm đảo chiều tính phân cực cho cả câu. Tuy nhiên, một từ phân cực về phía tiêu cực mạnh (điểm số rất thấp), không có nghĩa rằng sẽ phân cực về phía tích cực mạnh (điểm số rất cao). Nghiên cứu [31] sử dụng chiến lược *shift negation*. Thay vì chỉ đơn thuần đổi dấu điểm số, *shift negation* chỉ cộng/trừ 1 lượng cố định (trong nghiên cứu này là 4) vào điểm số của từ bị phủ định.

Ví dụ

`This CD is not horrid`
Điểm số của `horrid` là -5, khi đó, `not horrid` có điểm số là: -5 + 4 = -1

Tính điểm số cho câu Sau khi đã tính điểm cho các từ, SO-CAL tính điểm cho câu dựa trên nguyên tắc: Lấy trung bình cộng điểm số những từ có điểm số khác 0. Trường hợp tất cả các từ có điểm số bằng 0 thì điểm số của câu cũng bằng 0

5 Hiện thực hệ thống

Trong phần này, chúng tôi sẽ trình bày chi tiết về kỹ thuật cách rút trích các đặc trưng, hiện thực bộ phân loại cũng như tích hợp các yếu tố đó vào hệ thống. Ngôn ngữ lập trình được sử dụng là Python, phiên bản 2.7.

5.1 Thư viện và công cụ sử dụng

Chúng tôi dùng các thư viện hỗ trợ: UMLS-Metathesaurus, NLTK và Scikit-learn, cùng một số thư viện khác. Tất cả các thư viện đều là mã nguồn mở và miễn phí.

MetaMap

MetaMap cung cấp 3 phương pháp cơ bản để nhận diện các khái niệm trong kho dữ liệu UMLS-Metathesaurus từ dữ liệu đầu vào:

- Tương tác trực tiếp thông qua giao diện web[1] (*Use MetaMap Interactively*): MetaMap cung cấp giao diện web (Hình 10) giúp người dùng, đặc biệt là người mới sử dụng, có cái nhìn trực quan nhất về cách MetaMap hoạt động bao gồm cấu trúc dữ liệu đầu vào và đầu ra, những tùy chọn và cách các tùy chọn này ảnh hưởng đến kết quả phân tích. Tuy nhiên, mỗi lần gửi dữ liệu lên máy chủ MetaMap thông qua giao diện web này, MetaMap giới hạn dữ liệu đầu vào chỉ có một tập tin dữ liệu dạng text chứa không quá 10000 ký tự vì thế không thể sử dụng cách này cho những mẩu dữ liệu quá dài. Hơn nữa việc gửi và nhận kết quả trực tuyến thông qua mạng nên không đảm bảo tốc độ xử lý và bảo mật dữ liệu. Với những hạn chế vừa nêu, sau khi đã làm quen với cách hoạt động của MetaMap, người dùng có thể chọn 2 cách sau để tương tác với MetaMap.

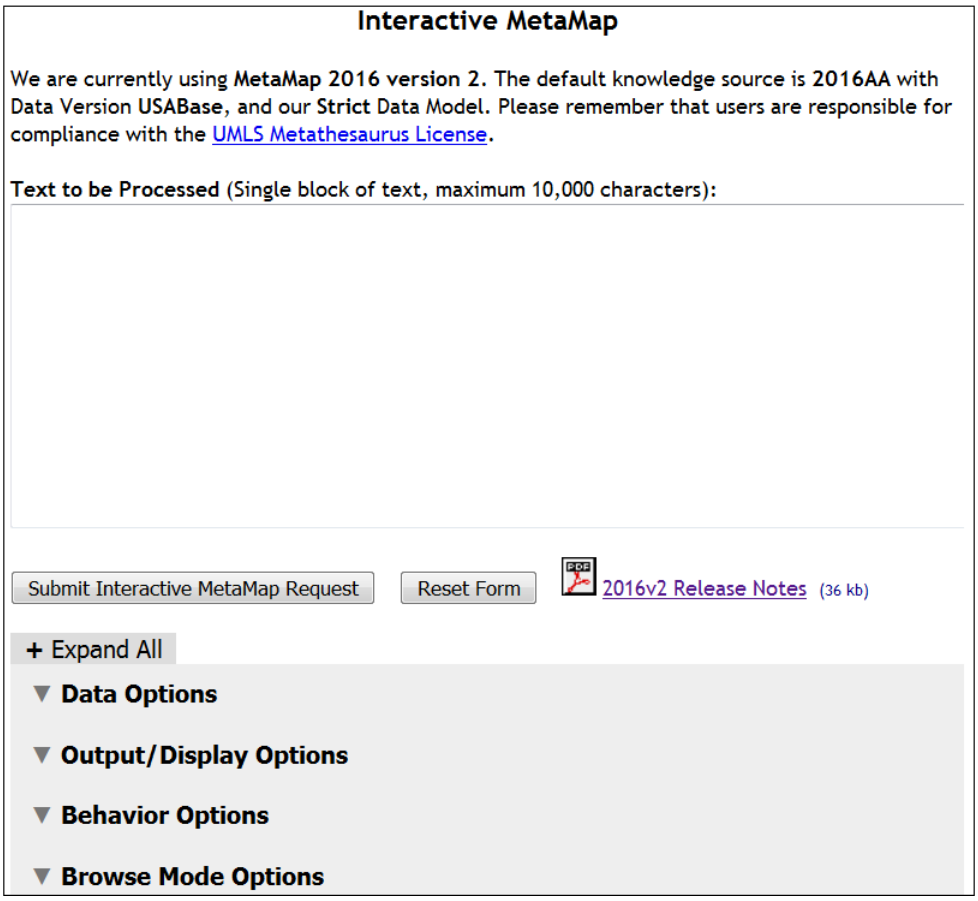
- Gửi dữ liệu lên máy chủ MetaMap[2] (*Use Batch MetaMap*): người dùng có thể gửi bộ dữ liệu lên máy chủ của MetaMap (Hình 11), sau quá trình xử lý MetaMap sẽ trả kết quả về địa chỉ email người dùng cung cấp. Ưu điểm của phương pháp này là tốc độ xử lý nhanh do không có tương tác trong quá trình phân tích dữ liệu. Tuy nhiên khi xảy ra lỗi trong quá trình chạy thì khó phán đoán được lỗi xảy ra ở mẩu dữ liệu nào.

- Sử dụng MetaMap cục bộ (*Use MetaMap Locally*): Nếu người dùng muốn hoàn toàn kiểm soát dữ liệu của mình thì việc cài đặt MetaMap ngay trên máy tính cá nhân là lựa chọn tốt nhất. Ưu điểm của phương pháp này là tốc độ xử lý nhanh, không phụ thuộc vào hệ thống mạng do không cần gửi dữ liệu lên máy chủ MetaMap, đảm bảo quyền kiểm soát dữ liệu và khả năng điều chỉnh cấu trúc kết quả phù hợp với nhu cầu sử dụng. Tuy nhiên nhược điểm lớn của phương pháp này là tốn khá nhiều tài nguyên hệ thống và cần máy tính có cấu hình đủ mạnh để làm máy chủ cục bộ.

Kết quả xử lý trả về như nhau khi dùng cả 3 phương pháp. Trong hệ thống phân loại chúng tôi đã dùng MetaMap cục bộ để tối đa khả năng điều chỉnh cấu trúc dữ liệu đầu ra.

Trong phạm vi luận văn, chúng tôi sử dụng công cụ MetaMap để xác định các nhãn y khoa tồn tại trong câu dữ liệu đầu vào. Chúng tôi đã tùy chọn bộ lọc nhãn từ vựng thuộc các nhóm ngữ nghĩa liên quan trực tiếp tới các triệu chứng, nguyên nhân, bệnh lý lâm

[1] https://ii.nlm.nih.gov/Interactive/UTS_Required/metamap.shtml
[2] https://ii.nlm.nih.gov/Batch/UTS_Required/metamap.shtml



Hình 10: Giao diện web tương tác trực tiếp của MetaMap

sàng và cận lâm sàng gồm *Pathologic Function*, *Disease or Syndrome*, *Mental or Behavioral Dysfunction*, *Cell or Molecular Dysfunction*, *Virus*, *Neoplastic Process*, *Anatomical Abnormality*, *Acquired Abnormality*, *Congenital Abnormality*, *Injury or Poisoning*.

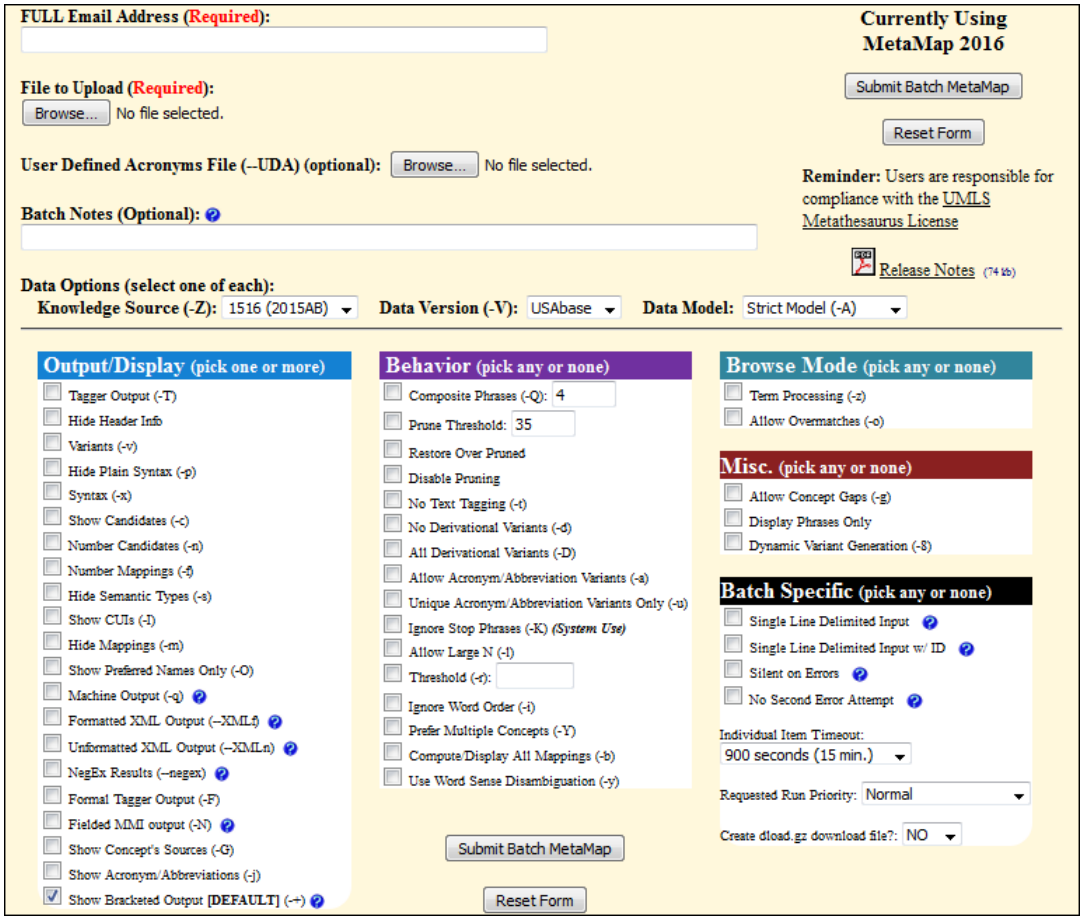
Trong quá trình sử dụng MetaMap, chúng tôi chỉnh tùy chọn `–negex` để lấy được kết quả đầu ra có phân tích phủ định. Đồng thời chúng tôi cũng xây dựng một chương trình con gọi là NegEx sử dụng giải thuật phân tích phủ định NegEx[1] để so sánh với kết quả từ MetaMap.

NLTK

NLTK [4] là một *platform* xây dựng trên ngôn ngữ lập trình Python, cung cấp các công cụ để thao tác với văn bản. NLTK tích hợp hơn 50 *corpora* và các nguồn từ vựng (*lexicon*) như WordNet, SentiWordNet cùng các thư viện để xử lý ngôn ngữ tự nhiên, hỗ trợ: *tokenization*, *stemming*, *lemmination*, *parsing*. Ngoài ra NLTK đóng gói và cung cấp các giao diện lập trình (*API*) của các thư viện khác (thư viện Stanford NLP).

Trong nghiên cứu này, chúng tôi sử dụng NLTK như công cụ chính để thực hiện các bước tiền xử lý: tách câu, *tokenization*, *stemming*, *lemmination*

[1]https://code.google.com/p/negex/wiki/NegExAlgorithmDescription



Hình 11: Batch MetaMap

Scikit-learn

Scikit-learn [25] là bộ thư viện về lĩnh vực Học máy trên nền tảng thư viện SciPy, sử dụng ngôn ngữ lập trình Python. Thư viện này cung cấp rất nhiều công cụ mà một bài toán Học máy cần dùng: Các giải thuật phân loại, hồi quy, phân cụm, . . . ; các công cụ tiền xử lý; các giải thuật thu giảm chiều; và các công cụ giúp lựa chọn, đánh giá mô hình.

Trong nghiên cứu này, chúng tôi sử dụng Scikit-learn là công cụ chính để hiện thực đặc trưng N-gram và giải thuật học máy SVM.

Các thư viện khác

Ngoài 2 thư viện kể trên, chúng tôi sử dụng thêm:

- NumPy giúp thao tác trên dữ liệu dạng vector.
- Pandas giúp đọc file, viết file và quản lý tập dữ liệu huấn luyện và dữ liệu để đánh giá.

5.2 Hiện thực rút trích đặc trưng

Trong phần này, chúng tôi sẽ trình bày cách thức hiện thực để từ một câu văn chuyển thành một vector, hoặc từ danh sách các câu chuyển thành một mảng 2 chiều. Từ đó làm input cho giải thuật học máy SVM

Đặc trưng N-gram

Scikit-learn cung cấp các công cụ để làm việc với dữ liệu dạng text. Trong số này, chúng tôi sử dụng module sklearn.feature_extraction.text.CountVectorizer để hiện thực đặc trưng N-gram. Như mô tả ở mục 4.3, n-gram được trích xuất qua 2 bước:

Bước đầu tiên: Xây dựng tập từ vựng T. Scikit-learn cung cấp lớp CountVectorizer cho tác vụ này.

```
vectorizer = CountVectorizer(input, min_df, binary, ngram_range)   1
```

Hàm khởi tạo CountVectorizer cung cấp 17 tham số, tuy nhiên, trong nghiên cứu này chúng tôi chỉ quan tâm đến 4 tham số, các tham số còn lại sử dụng giá trị mặc định:

input Là các câu trong tập dữ liệu huấn luyện

min_df Là số nguyên thể hiện số câu câu ít nhất chứa n-gram để n-gram đó được thêm vào tập từ vựng. Tham số này đã được giải thích chi tiết tại 4.3

binary Là giá trị True hoặc False. Nếu $binary = True$, giải thuật sử dụng *vector nhị phân*[1], ngược lại sử dụng *vector số nguyên*[2]

ngram_range Là một *tuple*, có dạng (a, b). Giải thuật sẽ sử dụng các loại n-gram từ a-gram đến b-gram.
Ví dụ $ngram\_range = (1, 3)$: Giải thuật sử dụng 3 loại n-gram: Uni-gram, bi-gram và tri-gram.

Bước thứ 2: xây dựng tập vector đại diện cho mỗi câu trong tập huấn luyện. Bước này được thực hiện bằng cách gọi hàm:

```
vectors-ngram = vectorizer.transform(raw_documents).toarray()   1
```

trong đó *raw_documents* là tập hợp các câu. Hàm trên trả về một mảng 2 chiều $n * m$ với n là số lượng câu trong tập *raw_documents* và m là kích thước của tập từ vựng

Đặc trưng Chang phrase

Đặc trưng Change phrase ánh xạ một câu sang 1 vector 4 chiều. Trong phần hiện thực, chúng tôi định nghĩa hàm training_change_phrase:

```
def training_change_phrase(raw_documents):   1
```

Hàm trả về mảng 2 chiều $n * 4$ với n là số lượng câu trong tập *raw_documents*

Thành phần phủ định

Với dữ liệu đầu vào là câu dạng ngôn ngữ tự nhiên, chúng tôi xây dựng 2 công cụ để xác định thành phần phủ định trong câu:

- Meta-NegEx: chỉnh tùy chọn

  -negex để lấy được kết quả phân tích của MetaMap có chứa thành phần phủ định và phạm vi phủ định. Kết quả lưu lại dưới dạng JSON <Thêm ví dụ>

[1] 2 khái niệm này được định nghĩa tại Mục 4.3, phần Rút trích
[2] 2 khái niệm này được định nghĩa tại Mục 4.3, phần Rút trích

Gen-NegEx: dựa vào giải thuật phân tích phủ định NegEx đã mô tả ở Mục 4.5, chúng tôi xây dựng chương trình với kết quả đầu ra là câu dữ liệu đã được xác định thành phần và phạm vi phủ định. <Thêm ví dụ>

Việc hiện thực song song hai công cụ phân tích phủ định nhằm so sánh hiệu quả của chúng và chọn ra công cụ phù hợp nhất.

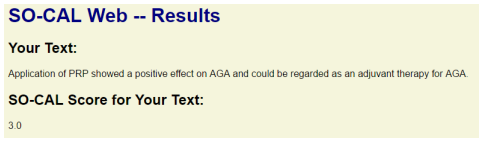
Với Meta-NegEx, công cụ hỗ trợ phiên bản local được cung cấp miễn phí. Một kết quả phân tích thử nghiệm như Hình
HÌNH

Với Gen-NegEx, chúng tôi sử dụng mã nguồn của/tại.. Tương tự, kết quả phân tích 1 câu như Hình
HÌNH
Kết quả của 2 cộng cụ được xử lý để đưa về chung 1 cấu trúc, sau đó lưu vào file. Việc này giúp dễ dàng tích hợp yếu tố vào hệ thống các đặc trưng mà không quan tâm tới quá trình trích xuất bằng công cụ nào.

SO-CAL

Trong phần này, chúng tôi sử dụng phiên bản hiện thực của nghiên cứu [31]. Nhóm tác giả của nghiên cứu [31] đã hiện thực công cụ online để tính điểm số cho 1 đoạn văn, giao diện được mô tả như Hình 12.

(A) Nhập 1 đoạn text sau đó nhấp submit để chuyển đến trang kết quả

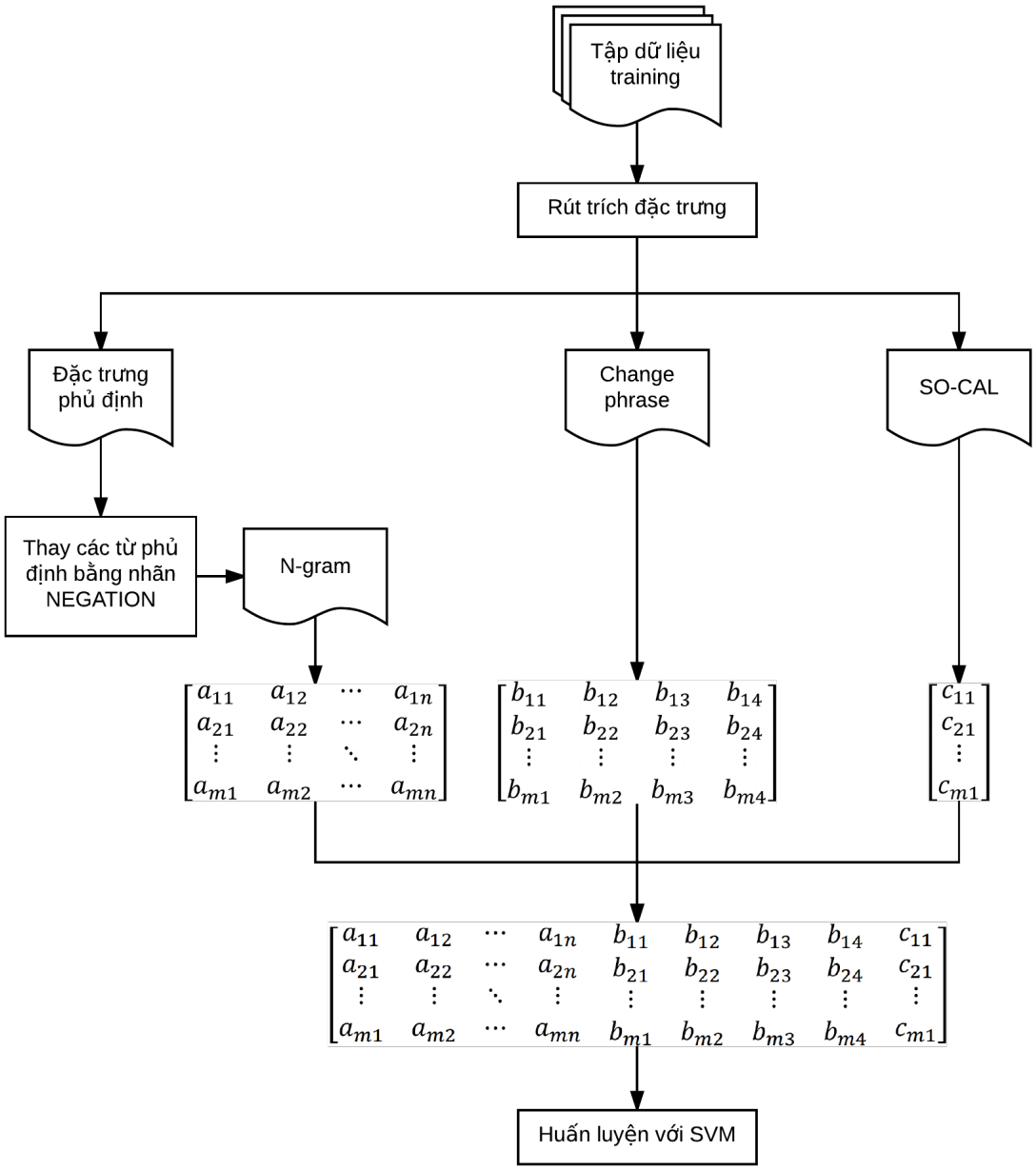


(B) Giao diện kết quả

HÌNH 12: Công cụ online để tính điểm SO-CAL

Để tích hợp vào hệ thống, nhóm sử dụng package `re` trong Python, sử dụng phương thức `HTTP POST` để gửi yêu cầu lên trang web, sau đó *parser* phần kết quả trả về để lấy ra thông tin điểm số.

5.3 Hiện thực bộ phân loại SVM



Hình 13: Kết hợp các đặc trưng trước khi đưa vào SVM huấn luyện

Các đặc trưng sau khi được rút trích như đã mô tả sẽ được kết hợp lại để với mỗi câu, chỉ có 1 vector đại diện. Mô hình kết hợp các đặc trưng được miêu tả như Hình 13
Chúng tôi sử dụng lớp SVM.SVC của thư viện Scikit-learn để hiện thực giải thuật học máy SVM.

```
clf = svm.SVC(decision_function_shape = 'ovr', C = c,                1
              kernel = 'rbf', class_weight = 'balanced')
clf.fit(data_x, data_y)                                              2
```

Hàm khởi tạo SVM.SVC ở dòng 1 có các tham số được sử dụng như sau:

decision_function_shape Định nghĩa cách SVC hiện thực bộ phân loại đa lớp. Nếu decision_function_shape='ovr' (one-vs-rest), SVC tạo ra 3 hàm quyết định (deci-

sion function): Một câu có thuộc lớp *Tích cực* hay không, một câu có thuộc lớp *Tiêu cực* hay không và một câu có thuộc lớp *Trung tính* hay không. Nếu decision_function_shape='ovo' (one-vs-one), SVC tạo ra 3 hàm quyết định: Một câu thuộc lớp *Tích cực* hay thuộc lớp *Tiêu cực*, một câu thuộc lớp *Tiêu cực* hay thuộc lớp *Trung tính* và một câu thuộc lớp *Trung tính* hay thuộc lớp *Tích cực*

C là hệ số được định nghĩa trong mô hình soft-margin của giải thuật SVM. Hệ số này đánh đổi giữa việc chấp nhận 1 vài dữ liệu bị phân loại sai, bù lại việc mặt phẳng hàm quyết định (decision surface) trở nên phức tạp, ít tuyến tính. C cao thì SVC càng phân loại chính xác các dữ liệu trong tập huấn luyện, ngược lại C thấp thì mặt phẳng hàm quyết định càng đơn giản.

kernel Định nghĩa loại *kernel* SVC sử dụng: linear, poly, rbf, sigmoid, precomputed

class_weight Tùy chọn cách xử lý khi số lượng các lớp trong tập huấn luyện bị chêch lệch

Sau khi khởi tạo, hàm clf.fit(data_x, data_y) được gọi với data_x là dữ liệu tập huấn luyện sau khi đã qua các bước rút trính đặc trưng để ánh xạ từ 1 câu sang 1 vector, data_y là vector chứa nhãn tương ứng.

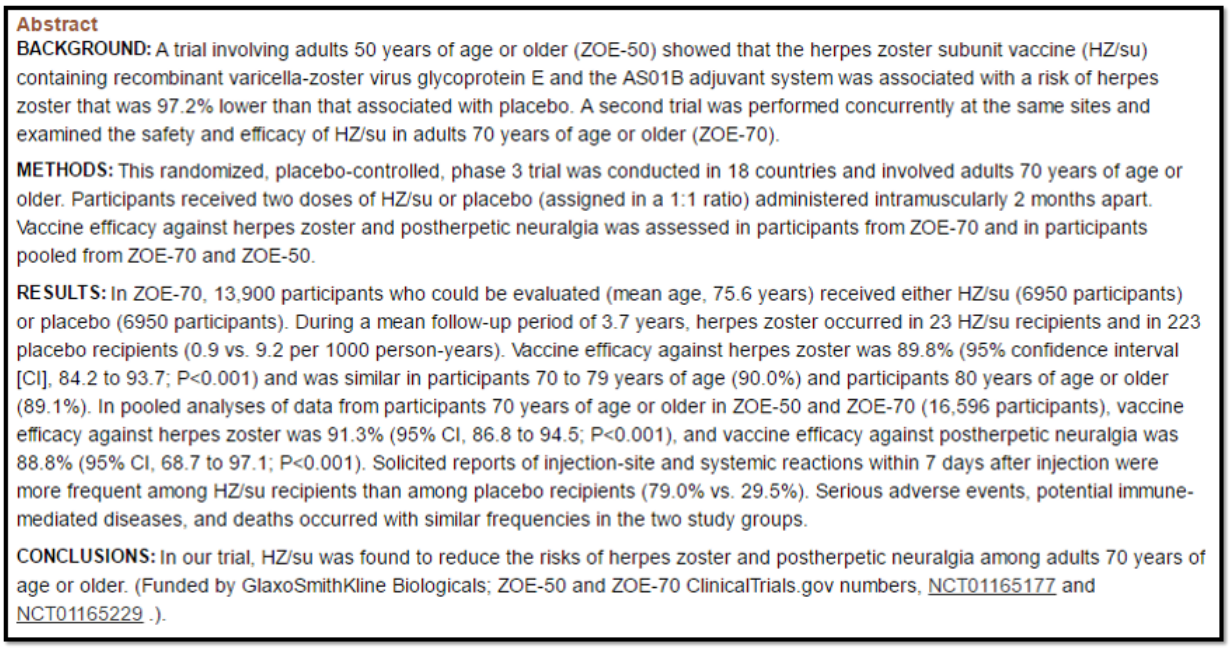
6 Thí nghiệm và đánh giá

Ở phần này chúng tôi sẽ trình bày chi tiết về tập dữ liệu đầu vào, các phương pháp đánh giá và kết quả thí nghiệm của hệ thống, cùng với những phân tích dựa trên kết quả thí nghiệm thu được.

6.1 Thu thập và đánh giá dữ liệu

Thu thập dữ liệu

Trong nghiên cứu này, chúng tôi thu thập dữ liệu từ trang web PubMed [1]. Trang web cung cấp miễn phí tóm tắt (*abstract*) của các bài báo khoa học trong lĩnh vực y khoa. Phần tóm tắt của các bài báo không có cấu trúc chung, chúng tôi chỉ thu thập những tóm tắt nào có cấu trúc các phần như Hình 14. Dựa theo báo cáo [22], chúng tôi chỉ giữ lại phần Kết luận, và sử dụng công cụ tìm kiếm cùng bộ lọc để tìm các bài có loại xuất bản (*publication type*) là "clinical trial". Dữ liệu chúng tôi thu thập được gồm 1182 câu được gán mã số (*Id*) tuần tự và lưu trữ như Bảng 6.



Abstract

BACKGROUND: A trial involving adults 50 years of age or older (ZOE-50) showed that the herpes zoster subunit vaccine (HZ/su) containing recombinant varicella-zoster virus glycoprotein E and the AS01B adjuvant system was associated with a risk of herpes zoster that was 97.2% lower than that associated with placebo. A second trial was performed concurrently at the same sites and examined the safety and efficacy of HZ/su in adults 70 years of age or older (ZOE-70).

METHODS: This randomized, placebo-controlled, phase 3 trial was conducted in 18 countries and involved adults 70 years of age or older. Participants received two doses of HZ/su or placebo (assigned in a 1:1 ratio) administered intramuscularly 2 months apart. Vaccine efficacy against herpes zoster and postherpetic neuralgia was assessed in participants from ZOE-70 and in participants pooled from ZOE-70 and ZOE-50.

RESULTS: In ZOE-70, 13,900 participants who could be evaluated (mean age, 75.6 years) received either HZ/su (6950 participants) or placebo (6950 participants). During a mean follow-up period of 3.7 years, herpes zoster occurred in 23 HZ/su recipients and in 223 placebo recipients (0.9 vs. 9.2 per 1000 person-years). Vaccine efficacy against herpes zoster was 89.8% (95% confidence interval [CI], 84.2 to 93.7; P<0.001) and was similar in participants 70 to 79 years of age (90.0%) and participants 80 years of age or older (89.1%). In pooled analyses of data from participants 70 years of age or older in ZOE-50 and ZOE-70 (16,596 participants), vaccine efficacy against herpes zoster was 91.3% (95% CI, 86.8 to 94.5; P<0.001), and vaccine efficacy against postherpetic neuralgia was 88.8% (95% CI, 68.7 to 97.1; P<0.001). Solicited reports of injection-site and systemic reactions within 7 days after injection were more frequent among HZ/su recipients than among placebo recipients (79.0% vs. 29.5%). Serious adverse events, potential immune-mediated diseases, and deaths occurred with similar frequencies in the two study groups.

CONCLUSIONS: In our trial, HZ/su was found to reduce the risks of herpes zoster and postherpetic neuralgia among adults 70 years of age or older. (Funded by GlaxoSmithKline Biologicals; ZOE-50 and ZOE-70 ClinicalTrials.gov numbers, NCT01165177 and NCT01165229 .).

Hình 14: Tóm tắt của 1 bài báo

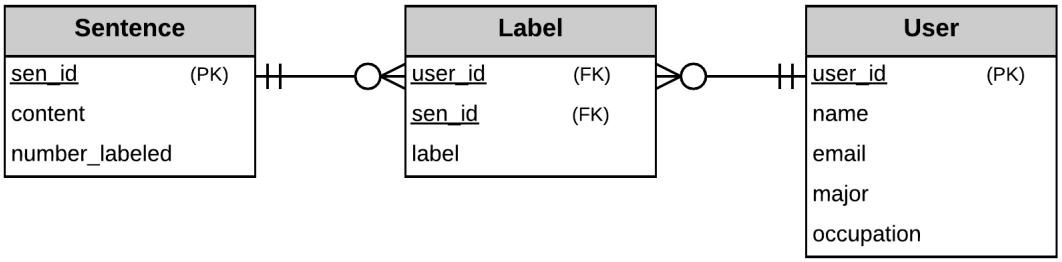
Bảng 6: Một số mẫu từ tập dữ liệu sau khi thu thập

Id	Sentence
10	This study was a negative study, though there was a suggestion of benefit of methylprednisolone acetate in a population of young adults with acute radicular low back pain.
17	Data extraction and analyses and quality assessment were conducted according to the Cochrane standards.
36	Patients subjectively reported significantly greater relief from symptoms with Debacterol than with Kenalog-in-Orabase or no treatment.

[1] https://www.ncbi.nlm.nih.gov/pubmed

Đánh nhãn dữ liệu

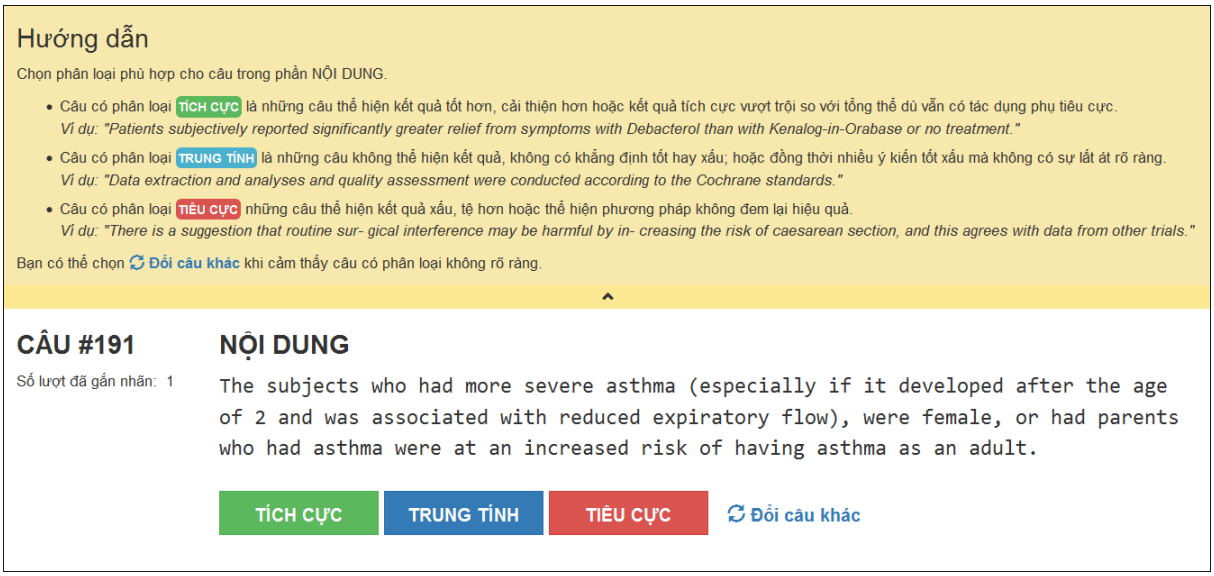
Giải thuật học máy chúng tôi sử dụng là có giám sát nên dữ liệu đầu vào cần được đánh nhãn phân loại tính phân cực cảm xúc (*Tích cực*, *Tiêu cực*, *Trung tính*) trước khi đưa vào học và kiểm tra. Vì vậy, chúng tôi đã hiện thực một trang web phục vụ cho việc đánh nhãn dữ liệu.



Hình 15: Mô hình thực thể liên kết tăng cường của cơ sở dữ liệu

Đầu tiên, chúng tôi sử dụng MySQL Workbench 6.3 [1] để tạo cơ sở dữ liệu lưu trữ theo lược đồ ở Hình 15:

- Bảng `Sentence` chứa dữ liệu các câu gồm mã số câu (`sen_id`), nội dung câu (`content`) và số lần câu được đánh nhãn (`number_labeled`). Tập dữ liệu sau khi thu thập sẽ được thêm vào bảng này với giá trị số lần câu được đánh nhãn ở mỗi câu ban đầu mặc định bằng 0.
- Bảng `Submission` chứa nhật ký đánh nhãn gồm mã số nhãn (`id`) tăng tuần tự theo mỗi nhãn được đánh cho một câu bất kỳ, mã số câu (`sentence_id`) tương ứng với mã số câu ở Bảng `Sentence`, loại nhãn (`label`) với quy ước giá trị 0, 1, 2 lần lượt tương ứng cho các nhãn *Tiêu cực*, *Trung tính*, *Tích cực*.

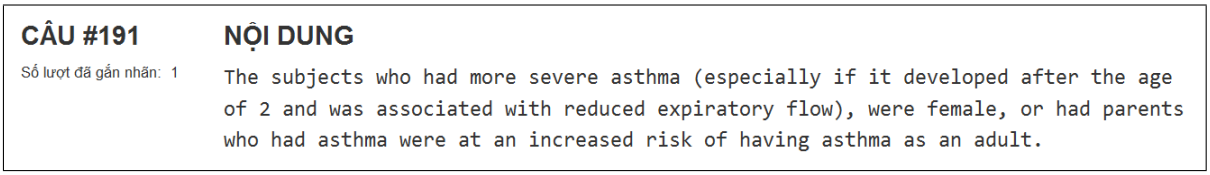


Hình 16: Giao diện trang đánh nhãn dữ liệu

[1] http://www.mysql.com/products/workbench/

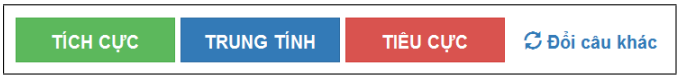
Tiếp theo, chúng tôi xây dựng một trang web hỗ trợ người dùng đánh nhãn dữ liệu[1]. Giao diện web đơn giản, trực quan (Hình 16), gồm hai phần chính:

- Phần hướng dẫn: mô tả cách sử dụng trang web để đánh nhãn dữ liệu, trong đó đặc tả chi tiết thế nào là phân loại *Tích cực*, *Tiêu cực* hay *Trung tính*, lấy ví dụ cụ thể để người đọc dễ hình dung và hiểu rõ ràng về các loại nhãn phân loại.
- Phần đánh nhãn: hiển thị thông tin một câu ngẫu nhiên thuộc bảng `Sentence` trong cơ sở dữ liệu và các lựa chọn để người dùng đánh nhãn phân loại.



HÌNH 17: Thông tin một bản ghi thuộc bảng `Sentence`

Thông tin chi tiết một câu được hiển thị như Hình 17 bao gồm giá trị các trường thuộc bảng `Sentence` (mã số câu, nội dung câu, số lần câu được đánh nhãn). Để gắn nhãn phân loại cho câu, người dùng sử dụng nhóm nút chức năng (Hình 18) gồm nút TÍCH CỰC (gắn nhãn *Tích cực*), nút TRUNG TÍNH (gắn nhãn *Trung tính*), nút TIÊU CỰC (gắn nhãn *Tiêu cực*) hoặc lựa chọn "Đổi câu khác" nếu người dùng cảm thấy câu có phân loại không rõ ràng.



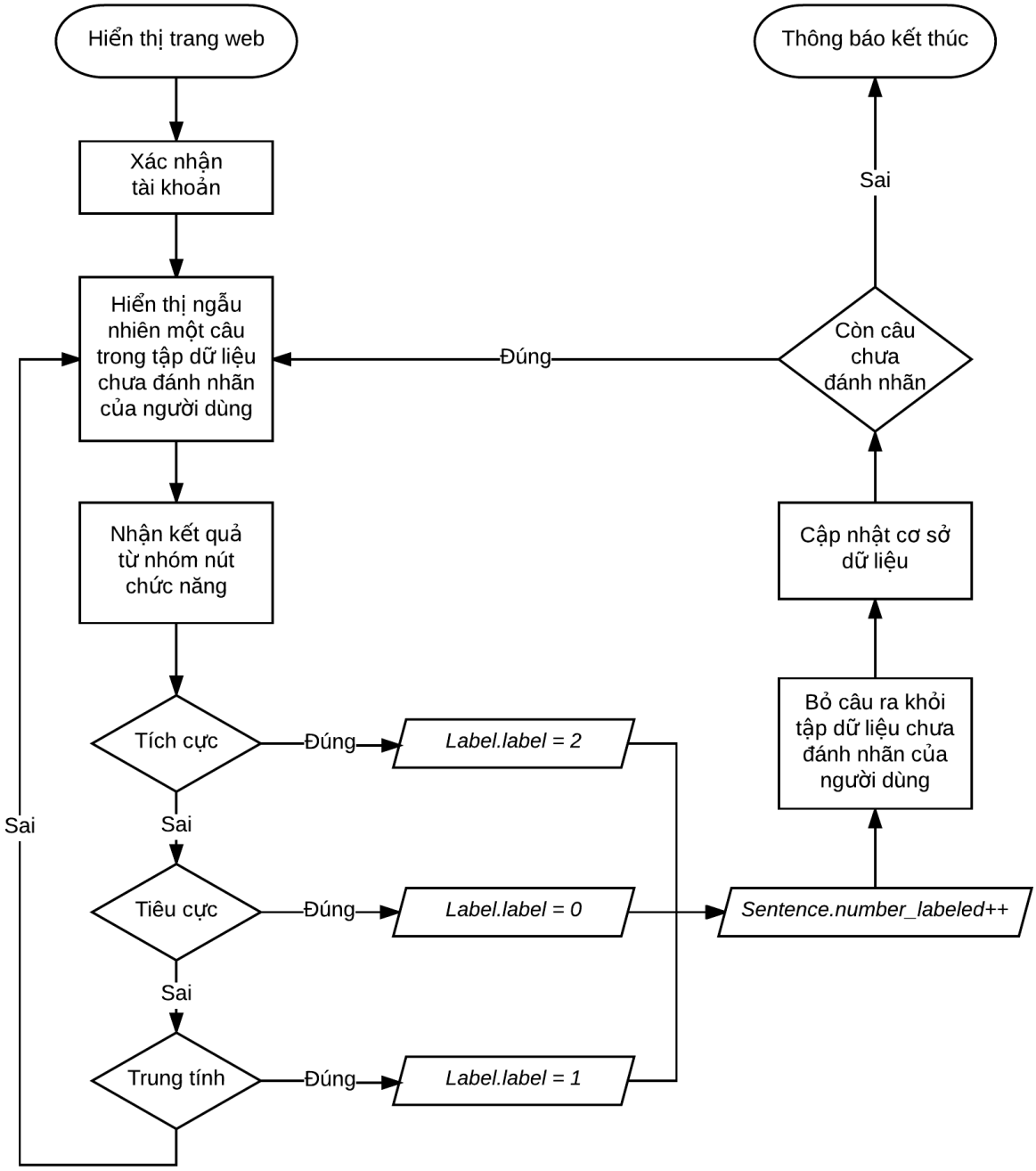
HÌNH 18: Nhóm nút chức năng hỗ trợ người dùng lựa chọn phân loại

Quy trình gắn nhãn dữ liệu của trang web được mô hình hóa như Hình 19. Mỗi lần người dùng mới truy cập hoặc tải lại trang web, hệ thống lựa chọn ngẫu nhiên một câu trong dữ liệu để hiển thị. Khi người dùng đánh nhãn *Tích cực*, *Tiêu cực* hoặc *Trung tính* cho câu, cơ sở dữ liệu sẽ cập nhật dữ liệu trong bảng `Sentence`: tăng thêm 1 cho số lần câu được đánh nhãn với mã số câu tương ứng, đồng thời thêm một bản ghi vào bảng `Submission` với mã số câu tương ứng kèm lựa chọn phân loại của người dùng (chỉ lưu các giá trị 0, 1, 2). Nếu người dùng chọn "Đổi câu khác" hệ thống không cập nhật mà chỉ hiển thị ngẫu nhiên một câu khác trong tập dữ liệu để người dùng tiến hành gán nhãn.
Sau khi hoàn thành trang web đánh nhãn dữ liệu, chúng tôi đã tiến hành gắn nhãn cho tập dữ liệu. Kết quả tập dữ liệu sau khi gắn nhãn được lưu lại như Bảng 7.

BẢNG 7: Một số mẫu từ tập dữ liệu sau khi đánh nhãn

Id	Sentence	Label
10	This study was a negative study, though there was a suggestion of benefit of methylprednisolone acetate in a population of young adults with acute radicular low back pain.	0
17	Data extraction and analyses and quality assessment were conducted according to the Cochrane standards.	1
36	Patients subjectively reported significantly greater relief from symptoms with Debacterol than with Kenalog-in-Orabase or no treatment.	2

[1]http://anotation.mybluemix.net/



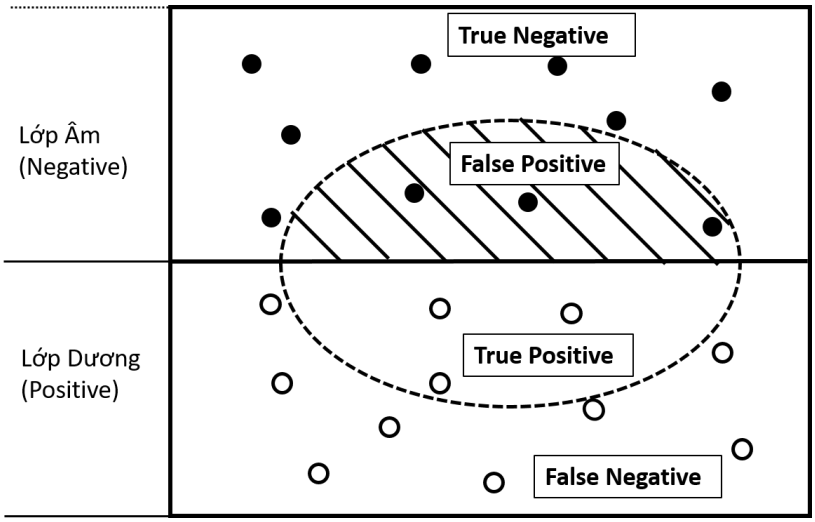
Hình 19: Quy trình xử lý gắn nhãn dữ liệu của trang web

Đánh giá dữ liệu

6.2 Phương pháp đánh giá

Sau khi huấn luyện hệ thống với tập dữ liệu *training*, bước tiếp theo là đánh giá kết quả huấn luyện để xác định mức độ tin cậy cũng như hiệu quả của hệ thống. Chúng tôi sử dụng 3 phép đo sau để đánh giá hiệu quả của hệ thống: Độ chính xác (Precision), độ bao phủ (Recall), và f. Đây cũng là các độ đo thường được sử dụng trong các bài toán về học máy và truy hồi thông tin. Cả 3 độ đo này chỉ được áp dụng trực tiếp vào bài toán phân loại nhị phân. Điều kiện trước tiên để áp dụng các độ đo này là cần quy ước 1 lớp (bài toán phân loại nhị phân chỉ có 2 lớp) là lớp dương (positive), là lớp mà hệ thống quan

tâm nhiều hơn. Ví dụ trong bài toán phân loại người bị ung thư với người không bị ung thư, chúng ta có 2 lớp: bị ung thư và không bị ung thư. Vậy để áp dụng 3 độ đo trên, người thiết kế hệ thống cần quy ước lớp bị ung thư là lớp dương đối với bài toán này (người thiết kế cũng có thể quy ước ngược lại, việc này phụ thuộc vào ngữ cảnh của bài toán: hệ thống cần chọn ra những người bị ung thư trong cộng đồng, hay cần chọn ra những người không bị ung thư trong cộng đồng.



Hình chữ nhật Toàn bộ dữ liệu
Hình eclip Phần dữ liệu hệ thống cho rằng thuộc lớp Positive
True Negative Phần dữ liệu thuộc lớp Negative, hệ thống cũng cho rằng thuộc lớp Negative
False Negative Phần dữ liệu thuộc lớp Positive, hệ thống cho rằng thuộc lớp Negative
True Positive Phần dữ liệu thuộc lớp Positive, hệ thống cũng cho rằng thuộc lớp Positive
False Positive Phần dữ liệu thuộc lớp Negative, hệ thống cho rằng thuộc lớp Positive

Hình 20: Các thành phần trong các phép đo Độ chính xác, Độ bao phủ và f1

Dựa trên câu trả lời là tập hợp các phần tử mà hệ thống cho rằng thuộc lớp dương, toàn bộ dữ liệu sẽ được chia thành 4 nhóm như Hình 20. Từ đó, định nghĩa các phép đo như sau:

Độ chính xác (Precision)

Độ chính xác P là hệ số đánh giá mức độ chính xác của câu trả lời, mức độ chính xác càng cao thì giá trị P càng lớn. Công thức tính P như sau:

$$P = \frac{\text{Tổng số câu trả lời đúng hệ thống đưa ra}}{\text{Tổng số câu trả hệ thống đưa ra}} = \frac{|\text{True Positive}|}{|\text{True Positive}| + |\text{False Positive}|}$$

Trong trường hợp tất cả những câu trả lời hệ thống đưa ra đều đúng, thì giá trị $P = 1$ là lớn nhất. Quy ước rằng nếu hệ thống không đưa ra câu trả lời nào, khi đó ngầm hiểu hệ thống không "sai", giá trị $P = 1$.

Độ bao phủ (Recall)

Độ bao phủ R là hệ số đánh giá mức độ bao phủ của các câu trả lời, độ bao phủ càng cao thì R càng lớn. Công thức tính R như sau:

$$R = \frac{\text{Tổng số câu trả lời đúng hệ thống đưa ra}}{\text{Tổng số câu trả lời đúng thực tế}} = \frac{|\text{True Positive}|}{|\text{True Positive}| + |\text{False Negative}|}$$

Trong trường hợp hệ thống đưa ra đủ tất cả các câu trả lời đúng thì giá trị $R = 1$ là lớn nhất.

F-measure

Trên thực tế, với lượng dữ liệu lớn và phức tạp, khả năng tất cả những câu trả lời hệ thống đưa ra đều đúng và hệ thống đưa ra đủ tất cả các câu trả lời đúng ($P = R = 1$) là rất thấp. Hai hệ số này thường bù trừ lẫn nhau, tức là để đạt độ bao phủ R cao, hệ thống có xu hướng đưa ra nhiều câu trả lời hơn làm cho xác suất có câu trả lời sai tăng, độ chính xác P giảm, và ngược lại. Một hệ thống nếu chỉ có P cao mà R thấp thì tuy hệ thống có câu trả lời thường đúng nhưng lại bỏ sót nhiều trường hợp đúng khác. Hệ thống chỉ có R cao mà P thấp thì hệ thống đó tuy bao quát đầy đủ tất cả các trường hợp đúng thực tế, nhưng tỉ lệ câu trả lời sai lại lớn. Một hệ thống tốt yêu cầu cả độ chính xác P và độ bao phủ đều cao R. Do đó vấn đề đặt ra là tìm một độ đo duy nhất mà đảm bảo cả P và R để thuận tiện cho việc tối ưu.

Để giải quyết vấn đề trên, ta sử dụng tiêu chí đánh giá F là trung bình điều hòa của P và R, từ đó đảm bảo rằng chỉ khi cả P và R cao thì F mới đạt giá trị cao. Công thức tính F như sau:

$$F = 2 * \frac{P * R}{P + R}$$

Độ chính xác (P), độ bao phủ (R), F đối với bài toán phân loại đa lớp

Để áp dụng 3 độ đo trên vào bài toán phân loại 3 lớp, chúng tôi xem như đang giải 3 bài toán con thuộc loại bài toán phân loại nhị phân. Khi đó, điểm số của bài toán lớn bằng trung bình có trọng số của điểm số từng bài toán con. Cụ thể: Xét 3 bài toán con:

1. Phân loại một câu thuộc lớp *Tích cực* hay không thuộc lớp *Tích cực*. Lớp dương được chọn là lớp *Tích cực*
2. Phân loại một câu thuộc lớp *Tiêu cực* hay không thuộc lớp *Tiêu cực*. Lớp dương được chọn là lớp *Tiêu cực*
3. Phân loại một câu thuộc lớp *Trung tính* hay không thuộc lớp *Trung tính*. Lớp dương được chọn là lớp *Trung tính*

Khi đó:

$$P = \alpha_1 * P_1 + \alpha_2 * P_2 + \alpha_3 * P_3$$

$$R = \alpha_1 * R_1 + \alpha_2 * R_2 + \alpha_3 * R_3$$

$$F = \alpha_1 * F_1 + \alpha_2 * F_2 + \alpha_3 * F_3$$

với $\alpha_1 + \alpha_2 + \alpha_3 = 1$ và $\alpha_1, \alpha_2, \alpha_3$ lần lược là tỉ lệ số lượng các câu thuộc lớp *Tích cực*, *Tiêu cực*, *Trung tính* trong tập dữ liệu huấn luyện.

K-fold cross validation

Sau khi đã có thước đo F, chúng tôi đề xuất sử dụng phương pháp *k-fold cross validation* để đánh giá hiệu quả của hệ thống. Phương pháp này nhằm tránh trường hợp tập kiểm tra, vì được chia ngẫu nhiên, có thể rơi vào trường hợp quá dễ hoặc quá khó đối với hệ thống. Tập dữ liệu được chia ngẫu nhiên thành k phần. Phần thứ i sẽ được chọn làm tập để đánh giá, $k - 1$ phần còn lại dùng cho việc học các tham số của mô hình. Tiến trình trên được thực hiện k lần với i chạy từ 1 đến k, giá trị trung bình là kết quả cuối cùng dùng để đánh giá hệ thống.

6.3 Kết quả thí nghiệm

Với mỗi lần chạy, chúng tôi sử dụng k-fold với $k = 5$. Tuy nhiên, do tập dữ liệu không đủ lớn, kết quả các lần chạy có sự dao động, vì vậy, với mỗi lần thử nghiệm, chúng tôi lặp lại việc chạy k-fold 30 lần, sau đó lấy điểm số trung bình xem như kết quả cuối cùng. Đối với giải thuật học máy SVM, có 2 tham số cần được tùy chỉnh thích hợp. Để xác định tham số c, hệ thống chạy nhiều lần với c thuộc 1 khoảng cho trước (trong nghiên cứu này, c chạy từ 20 đến 30), từ đó chọn ra giá trị c tương ứng với độ đo f lớn nhất. Như vậy, với mỗi thử nghiệm, giá trị độ do f luôn là giá trị cao nhất được chọn từ những lần chạy với giá trị c thay đổi từ 20 đến 30.

Trong mục này chúng tôi trình bày 4 nhóm thử nghiệm: Thử nghiệm với đặc trưng n-gram, thử nghiệm với đặc trưng phủ định, thử nghiệm kết hợp các đặc trưng cơ bản và thử nghiệm kết hợp các đặc trưng cơ bản với đặc trưng mở rộng SO-CAL

Thử nghiệm với đặc trưng n-gram

Chúng tôi tiến hành thử nghiệm với đặc trưng n-gram trước tiên vì như đã phân tích ở mục **??**, n-gram được xem như đặc trưng nền tảng (baseline). Kết quả thử nghiệm thể hiện ở Bảng 8

Bảng 8: Các thử nghiệm nhằm tối ưu hóa đặc trưng n-gram

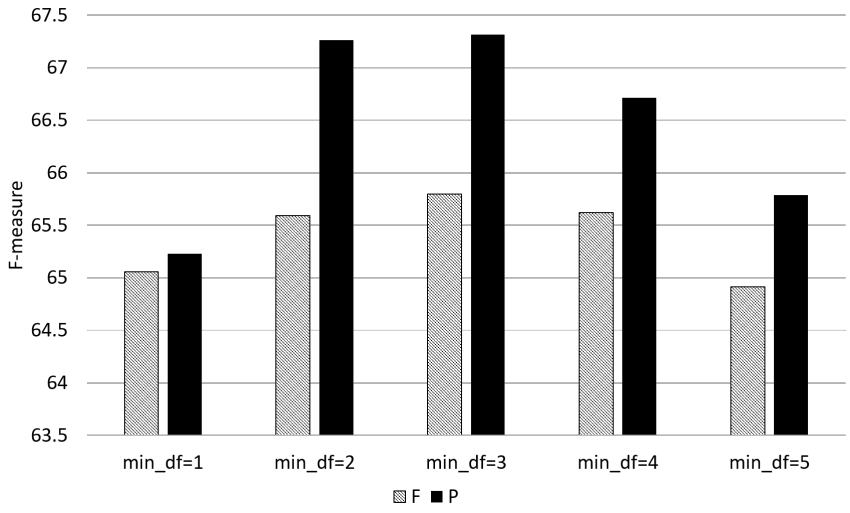
STT	**Đặc trưng**	**P (%)**	**R (%)**	**F (%)**
1	Unigram (F, min_df = 1)	66.21	65.46	65.06
2	Unigram (P, min_df = 1)	66.56	65.62	65.23
3	Unigram (F, min_df = 2)	66.50	65.55	65.59
4	Unigram (P, min_df = 2)	68.26	67.38	67.31
5	Unigram (F, min_df = 3)	66.98	65.48	65.80
6	Unigram (P, min_df = 3)	68.12	67.23	67.32
7	Unigram (F, min_df = 4)	67.02	65.13	65.62
8	Unigram (P, min_df = 4)	67.83	66.34	66.71
9	Unigram (F, min_df = 5)	66.75	64.21	64.92
10	Unigram (P, min_df = 5)	67.18	65.27	65.79
11	Unigram + Bigram (P, min_df = 3)	68.72	67.83	67.77
12	Unigram + Bigram + Trigram (P, min_df = 3)	68.68	68.00	67.87
13	**Unigram + Bigram + Trigram + 4-gram(P, min_df = 3)**	**68.76**	**68.09**	**67.96**
14	Unigram + Bigram + Trigram + 4-gram + 5-gram (P, min_df = 3)	68.81	67.98	67.86

P Chỉ quan tâm đến việc có xuất hiện hay không n-gram trong câu, nhận 2 giá trị: 1 hoặc 0
F Quan tâm đến số lần xuất hiện n-gram trong câu
min_df Số câu có n-gram đó để n-gram đó được thêm vào bộ từ vừng

Các thử nghiệm từ 1-10 cho thấy 2 xu hướng. Hình 21 thể hiện sự phụ thuộc của độ đo F vào tham số *min_df* và cách vector hóa đặc trưng n-gram. *min_df* là tham số ngưỡng, chỉ những n-gram nào có số lần xuất hiện từ *min_df* trở lên mới được thêm vào tập từ

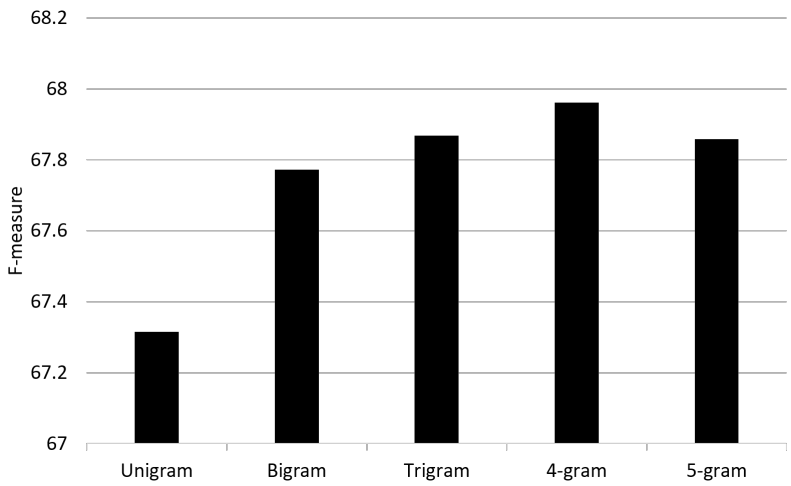
vựng S. Qua biểu đồ có thể thấy, với *min_df* quá nhỏ hoặc quá lớn đều làm giảm giá trị độ đo F. Khi *min_df* quá nhỏ, số lượng từ vựng quá lớn dẫn tới số lượng ngram gây nhiễu nhiều. Ngược lại khi *min_df* quá lớn, tập từ vựng quá nhỏ dẫn tới có quá ít thông tin trong câu được giữ lại, không đủ thông tin để phân loại. Từ biểu đồ, $min\_df = 2$ hay $min\_df = 3$ không có sự khác biệt rõ rệt giá trị độ đo F. Trong nghiên cứu này, chúng tôi chọn tham số $min\_df = 3$ cho các thử nghiệm còn lại.

Hình 21 còn thể hiện một xu hướng khác. Kết quả của n-gram tốt hơn hẳn khi sử dụng phương pháp vector hóa nhị phân. Kết quả này phù hợp với báo cáo của [24]. Ngược lại, nghiên cứu [28] thực hiện phân tích cảm xúc trên đoạn văn bản, khẳng định không có sự khác biệt đáng kể giữa 2 cách vector hóa. Điều này có thể do việc lặp từ trên câu có ý nghĩa khác với lặp từ trên đoạn.



Hình 21: Mối quan hệ giữa tham số min_df, cách vector hóa và độ đo F

Hình 22 thể hiện ảnh hưởng của cách kết hợp các n-gram. Theo đó, có sự cải thiện khi chuyển từ việc chỉ dùng Unigram sang dùng kết hợp Unigram và Bigram. Khi mở rộng việc kết hợp đến $n = 3$ và $n = 4$, mặc dù chỉ số F có tăng không thực sự đáng kể. Khi kết hợp đến $n = 5$, chỉ số F bắt đầu giảm. Với n càng lớn, mặc dù số lượng ngram không khác nhau nhiều (giả sử 1 câu có 20 từ, với $n = 1$ tạo ra 20 ngram, $n = 3$ tạo ra 18 ngram) nhưng các n-gram có tần suất xuất hiện càng thấp. Trong khi đó, hệ thống chỉ thêm ngram vào tập từ vựng S chỉ khi từ đó xuất hiện từ 3 lần trở lên. Vì vậy, việc kết hợp với các ngram (n lớn, $n = 5$ chẳng hạn) không giúp bổ sung thêm vào tập từ vựng S. Ngược lại, với $n = 2, 3, 4$ giúp hệ thống nhận thêm các cụm từ như: no evidence, improve quality life, reduce risk,... Trong các thử nghiệm tiếp theo, chúng tôi dùng đặc trưng n-gram là sự kết hợp của Unigram, Bigram, Trigram và 4-gram (ngram với $n = 4$).



Tên mỗi cột chỉ là đặc trưng ngram đại diện. Ví dụ: Trigram đại diện cho thử nghiệm thứ 12, là kết hợp cả Unigram, Bigram và Trigram

Hình 22: Kết hợp các n-gram

Thử nghiệm với đặc trưng Phủ định

Các thử nghiệm trong phần này đều dùng kết hợp với đặc trưng n-gram. Bảng 9 thể hiện hiệu quả của việc rút trích yếu tố phủ định qua 7 cách khác nhau, so sánh giữa 2 công cụ: Metamap sử dụng giải thuật NegEx và bản gốc NegEx. Trong 7 cách tích hợp yếu tố phủ định vào hệ thống, cách dùng của thí nghiệm 2 cho kết quả tốt nhất, xét cho cả 2 công cụ. Trong đó, NegEx bản gốc cho kết quả cao hơn. Chúng tôi sử dụng cách dùng như thí nghiệm 2 và công cụ NegEx bản gốc cho thí nghiệm ở phần sau.

Bảng 9: Các thử nghiệm nhằm tối ưu đặc trưng Phủ định

STT	**Đặc trưng**	**Negation tu metamap**			**Negation 2**		
		P	**R**	**F**	**P**	**R**	**F**
1	Kiểm tra trong câu có yếu tố phủ định hay không	69.36	68.53	68.45	69.59	68.32	68.50
2	**Thay các từ phủ định trong câu bằng nhãn NEGATION**	**69.57**	**68.67**	**68.60**	**70.07**	**68.98**	**69.09**
3	Thêm nhãn "_NEG" ngay sau các từ chịu ảnh hưởng phủ định	68.93	67.91	67.88	69.11	68.12	68.03
4	Kết hợp 1 và 2	68.88	68.10	68.09	69.95	68.64	68.82
5	Kết hợp 1 và 3	69.41	68.55	68.56	69.47	67.85	68.16
6	Kết hợp 2 và 3	68.84	68.02	67.93	69.00	68.00	67.83
7	Kết hợp 1 và 2 và 3	69.35	68.40	68.44	69.62	67.96	68.22

Thử nghiệm kết hợp các đặc trưng cơ bản

Bảng 10: Các thử nghiệm kết hợp các đặc trưng cơ bản

STT	Đặc trưng	P	R	F
1	N-gram	68.76	68.09	67.96
2	N-gram + *Change phrase*	70.15	69.42	69.35
3	**N-gram + *Change phrase* + Phủ định**	**71.09**	**70.11**	**70.11**
4	N-gram + *Change phrase* + Phủ định + Metamap	70.95	69.99	70.01

Bảng 10 thể hiện kết quả khi kết hợp các đặc trưng cơ bản với nhau. *Change phrase* có hiệu quả rõ rệt khi giúp tăng 1.39% so với *baseline* n-gram. Đặc trưng Phủ định cũng góp phần cải thiện kết quả. Kết quả khi sử dụng n-gram với Metamap lại không cho kết quả tốt. Điều này trái ngược với kết quả của nghiên cứu [28] và [22].

Thử nghiệm kết hợp các đặc trưng cơ bản với đặc trưng mở rộng SO-CAL

Bảng 11: Các thử nghiệm kết hợp các đặc trưng cơ bản với đặc trưng mở rộng SO-CAL

STT	Đặc trưng	P	R	F
1	N-gram + SO-CAL	23.45%	23.45%	23.45%
3	N-gram + Change_Phrase + SO-CAL	23.45%	23.45%	23.45%
4	N-gram + Change_Phrase + Negation + SO-CAL	23.45%	23.45%	23.45%
5	N-gram + Change_Phrase + Negation + UMLS + SO-CAL	23.45%	23.45%	23.45%

6.4 Các phân tích mở rộng

Sự phụ thuộc hiệu quả đặc trưng ngram với kích thước tập huấn luyện

So sánh kết quả đối với từng lớp

7 Tổng kết

7.1 Kết quả đạt được

7.2 Hạn chế và hướng phát triển

Tài liệu tham khảo

[1] Tanveer Ali et al. "Can I Hear You? Sentiment Analysis on Medical Forums.". In: *International Joint Conference on Natural Language Processing* (2013), pp. 667–673.

[2] Alan R Aronson and François-Michel Lang. "An overview of MetaMap: historical perspective and recent advances". In: *Journal of the American Medical Informatics Association* (2010).

[3] Farah Benamara et al. "How do Negation and Modality Impact on Opinions?". In: (2012), pp. 10–18.

[4] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[5] Olivier Bodenreider. "The Unified Medical Language System (UMLS): integrating biomedical terminology.". In: *Nucleic acids research* 32.Database issue (2004), pp. D267–70. ISSN: 1362-4962.

[6] S Chandrakala and C Sindhu. "Opinion Mining and Sentiment Classification a Survey". In: *Information and Communications Technology Academy of Tamil Nadu Journal on Soft Computing* 3.1 (2012), pp. 420–425.

[7] W W Chapman et al. "Evaluation of negation phrases in narrative clinical reports.". In: *Proceedings / AMIA ... Annual Symposium. AMIA Symposium* (2001), pp. 105–9. ISSN: 1531-605X.

[8] Wendy W Chapman et al. "Extending the NegEx lexicon for multiple languages.". In: *Studies in health technology and informatics* 192 (2013), pp. 677–81. ISSN: 0926-9630.

[9] Roberto Costumero et al. "An approach to detect negation on medical documents in Spanish". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8609 LNAI. Springer International Publishing, 2014, pp. 366–375.

[10] I G Councill, Ryan McDonald, and Leonid Velikovich. "What's great and what's not: learning to classify the scope of negation for improved sentiment analysis". In: *Proceedings of the ACL Workshop on Negation and Speculation in Natural Language Processing Uppsala Sweden* July (2010), pp. 51–59.

[11] Noa P Cruz Díaz and Manuel De Buenaga. "Negation and Speculation Detection in Clinical and Review Texts Detección de la Negación y la Especulación en Textos Médicos y de Opinión". In: *Procesamiento del Lenguaje Natural* (2015).

[12] Kerstin Denecke. "Sentiment Analysis from Medical Texts". In: *Health Web Science* (2015), pp. 75–81.

[13] Peter L Elkin et al. "A controlled trial of automated classification of negation from clinical notes.". In: *BMC medical informatics and decision making* 5.1 (2005), p. 13. ISSN: 1472-6947.

[14] Anastasia Giachanou and Fabio Crestani. "Like it or not: A survey of Twitter sentiment analysis methods". In: *ACM Comput Surv* 49.2 (2016), Article 28; 1–41. ISSN: 0360-0300.

[15] S Gindl. "Negation Detection in Automated Medical Applications: A Survey". In: Asgaard-{TR-2006-1} (2006), pp. 1–28.

[16] T. Givón. "English Grammar". In: Amsterdam: John Benjamins Publishing Company, 1993. ISBN: 978 90 272 2098 1.

[17] Thorsten Joachims. "Text categorization with support vector machines: Learning with many relevant features". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (1998), pp. 137–142.

[18] Tibor Kiss and Jan Strunk. "Unsupervised multilingual sentence boundary detection". In: *Computational Linguistics* 32.4 (2006), pp. 485–525.

[19] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. "An Introdution to Information Retrieval". In: *Journal Information Retrieval* (2009), pp. 319–348.

[20] P G Mutalik, A Deshpande, and P M Nadkarni. "Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS". In: *Journal of the American Medical Informatics Association* 8.6 (2001), pp. 598–609. ISSN: 1067-5027.

[21] Yun Niu, Xiaodan Zhu, and Graeme Hirst. "Using Outcome Polarity in Sentence Extraction for Medical Question-Answering". In: *Proceedings of the American Medical Informatics Association Symposium* (2006), pp. 599–603.

[22] Yun Niu et al. "Analysis of Polarity Information in Medical Text". In: *Proceedings of the American Medical Informatics Association Symposium* (2005), pp. 570–574.

[23] Bruno Ohana and Brendan Tierney. "Sentiment classification of reviews using SentiWordNet". In: *9th. IT & T Conference*. 2009, p. 13.

[24] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs Up?: Sentiment Classification Using Machine Learning Techniques". In: *Proceedings of the 2nd Association for Computational Linguistics Conference on Empirical methods in Natural Language Processing* 10 (2002), pp. 79–86.

[25] F Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[26] Ling Pei et al. "Using LS-SVM based motion recognition for smartphone indoor wireless positioning.". In: *Sensors (Basel, Switzerland)* 12.5 (2012), pp. 6155–75. ISSN: 1424-8220.

[27] John Pestian et al. "Sentiment Analysis of Suicide Notes: A Shared Task". In: *Biomedical Informatics Insights* 5 (2012), pp. 3–16. ISSN: 1178-2226.

[28] Abeed Sarker et al. "Outcome Polarity Identification of Medical Papers". In: *Proceedings of Australasian Language Technology Association Workshop* (2011), pp. 105–144.

[29] Maria Skeppstedt, Carita Paradis, and Andreas Kerren. "Marker words for negation and speculation in health records and consumer reviews". In: *Proceedings of the 7th International Symposium on Semantic Mining in Biomedicine* 1650 (2016), pp. 64–69.

[30] Phillip Smith and Mark Lee. "Cross-discourse development of supervised sentiment analysis in the clinical domain". In: *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*. Association for Computational Linguistics. 2012, pp. 79–83.

[31] Maite Taboada et al. "Lexicon-Based Methods for Sentiment Analysis". In: *Association for Computational Linguistics* 37.2 (2011), pp. 267–307.

[32] Hideyuki Tanushi et al. "Negation Scope Delimitation in Clinical Text Using Three Approaches: NegEx, PyConTextNLP and SynNeg". In: (2013).

[33] Anthony J Viera and Joanne M Garrett. "Understanding Interobserver Agreement : The Kappa Statistic". In: May (2005), pp. 360–363.

[34] Lei Xia, Anna Lisa Gentile, and James Munro. "Improving Patient OpinionMining through Multi-step Classification". In: *Proceedings of the 12th International Conference on Text, Speech and Dialogue* (2009), pp. 70–76.

[35] Qing Zeng et al. "Negation Detection using Regular Expression, Syntactic and Classification Methods". In: (2007), pp. 1–6.

[36] Lei Zhang et al. "Combining lexicon based and learning-based methods for twitter sentiment analysis". In: *HP Laboratories, Technical ...* (2011), p. 9477. ISSN: 2277-9477.

PHỤ LỤC (nếu có)