NGHIÊN CỨU CÁC MÔ HÌNH PHÂN LOẠI VĂN BẢN ĐỂ XÂY DỰNG CHATBOT TƯ VẤN TUYỂN SINH

A RESEARCH ON TEXT CLASSIFIERS TO BUILD A COLLEGE ADMISSION CHATBOT

Nguyễn Trí Bằng[1], Phan Trần Đăng Khoa[1], Tôn Quang Hoàng Nguyên[2]

[1]Trường Đại học Bách khoa - Đại học Đà Nẵng; ntbang@dut.udn.vn, ptdkhoa@dut.udn.vn
[2]Công ty Orient Software Đà Nẵng; nguyen.ton@orientsoftware.com

Tóm tắt - Trong bài toán phân loại văn bản, hầu hết các nghiên cứu trước đây đều so sánh đánh giá các mô hình huấn luyện trên một tập kiểm thử với kích thước nhất định, cũng như chưa làm rõ thời gian huấn luyện của các mô hình. Nghiên cứu này tập trung đánh giá độ chính xác của 3 mô hình phân loại văn bản: Support Vector Machine, Linear Regression, Naïve Bayes trên các tập kiểm thử với kích thước khác nhau; Sau đó nêu rõ các thông số đánh giá của mô hình với một tập kiểm thử có kích thước 900 câu hỏi. Bên cạnh đó, thời gian huấn luyện của từng mô hình cũng được so sánh trên các tập huấn luyện có kích thước khác nhau. Kết quả chỉ ra Naïve Bayes đều có độ chính xác tốt và thời gian huấn luyện nhanh nỗi trội so với 2 mô hình còn lại. Sau cùng, tác giả vận dụng kết quả nghiên cứu đề xuất giải pháp xây dựng một chatbot tư vấn tuyển sinh qua Facebook, cho kết quả thực nghiệm tốt và có thể ứng dụng tại các đơn vị giáo dục Việt Nam.

Từ khóa - phân loại văn bản, support vector machine, naïve bayes, linear regression, Facebook chatbot

Abstract - In text classification field, most of previous studies only measured and evaluated the selected model upon one test set with a certain size, as well as did not clarify the training time of each model. This paper focuses on comparing and evaluating 3 text classification models: Support Vector Machine, Linear Regression using SGD, Naïve Bayes in terms of accuracy with different test sets; Then clarify the evaluation parameters with a test set of 900 input questions. Besides, the reseach compares the training time of each model at different training sets with varied training-sizes. The results showed that Naïve Bayes has good accuracy and the training time is also significantly dominant when compared with two others. Afterall, the author uses the above research results to propose a solution to build an admissions chatbot through Facebook, provide promising empirical results and make it applicable to educational units in Vietnam.

Key words - text classifier, support vector machine, naïve bayes, linear regression, Facebook chatbot

1. Giới thiệu

Chatbot là một chương trình máy tính tương tác với người dùng bằng ngôn ngữ tự nhiên dưới một giao diện đơn giản, âm thanh hoặc dưới dạng tin nhắn. Theo Hakan Sundblad [1], phân lớp câu hỏi là nhiệm vụ gán một giá trị kiểu boolean cho mỗi cặp $(q_j, c_i) \in QxC$; trong đó: Q là miền chứa các câu hỏi, $C = \{c_1, c_2, \ldots, c_{|c|}\}$ là tập các phân lớp cho trước. Bài toán phân lớp (phân loại) câu hỏi có thể được hiểu như sau: "Với đầu vào gồm một tập các câu hỏi: $Q = \{q_1, q_2, \ldots, q_n\}$ và tập các lớp được định nghĩa: $C = \{c_1, c_2, \ldots, c_n\}$, thì đầu ra là nhãn c_i của câu hỏi q_j". Lúc này, chatbot phải có khả năng phân loại câu hỏi q_j để xem nó thuộc lớp c_i (ở đây được hiểu là lớp chủ đề) nào.

Các ứng dụng chatbot hiện nay hầu hết là ở trong lĩnh vực giáo dục [2] và đóng vai trò to lớn trong nền giáo dục tương lai [3]. Theo [4], 94% người dùng Internet tại Việt Nam sử dụng ứng dụng tin nhắn, 42% trong số đó cài đặt Facebook Messenger để nhắn tin. Nhu cầu tư vấn trực tuyến của phụ huynh học sinh về các ngành học ngày càng cao thì chatbot tư vấn tuyển sinh sẽ đáp ứng nhu cầu này một cách hiệu quả nhất. Khi mà nội dung câu hỏi của phụ huynh và các em học sinh đặt ra có tính chất lặp đi lặp lại, chatbot sẽ giúp đội ngũ tư vấn tiết kiệm công sức và thời gian đáng kể.

Có nhiều cách để phân loại nhưng dựa theo mô hình hoạt động, có thể phân chatbot thành 2 loại: rule-based chatbot và AI-based chatbot [5]. Chatbot trên Facebook đang được hỗ trợ phát triển dưới nhiều nền tảng và hoạt động theo nhiều mô hình khác nhau. Theo nghiên cứu [6] làm về một rule-based chatbot tư vấn sinh viên trên Facebook Messenger, việc cập nhật và chỉnh sửa các quy tắc là cực kỳ khó khăn; Khi mà các quy tắc ngày càng phức tạp và số lượng quy tắc ngày càng nhiều hơn. Theo [7], rule-based là phương pháp rất phổ biến, cho độ chính xác cao đối với miền câu hỏi phạm vi hẹp; nhưng ngược lại, nó tốn thời gian để xây dựng các quy tắc cũng như khó có thể kiểm soát vấn đề nảy sinh về sau. Nhằm giúp chatbot có thể tư vấn giống con người hơn, nhiều mô hình thuật toán phân loại văn bản đã được nghiên cứu và ứng dụng thành công trên chatbot. Ở trong [4], nhóm tác giả đã đề xuất giải pháp xây dựng chatbot phục vụ trong lĩnh vực giáo dục hoạt động trên Facebook Messenger sử dụng mô hình Naive Bayes mà về bản chất là phân loại nhóm từ khóa.

Theo nghiên cứu của Yiming Yang & Xin Liu [8] và ở cả trong [9], [10] thì Support Vector Machine, Naïve Bayes là 2 trong nhiều các mô hình phân loại văn bản phổ biến nhất hiện nay. Cũng có nghiên cứu về phân loại văn bản dựa vào mô hình phân loại theo chủ đề từ ngữ ứng dụng lí thuyết Naïve Bayes [11]. Naïve Bayes được đánh giá là có hiệu năng tốt [12] [13] cũng như dễ triển khai thực hiện [14] so với nhiều mô hình khác trong tác vụ phân loại văn bản. Dù hiệu năng chưa thực sự vượt trội [8], nhưng việc huấn luyện và kiểm thử của mô hình Naive Bayes là rất nhanh [14], [15]. Trong [16], [17], tác giả đã sử dụng phương pháp SGD để giải quyết bài toán phân loại văn bản; trong [18], tác giả sử dụng mô hình Linear Regression với phương pháp SGD nhằm phân tích dự đoán dữ liệu.

Ở trong [19], để chọn ra thuật toán phân loại lớp chủ đề trong chatbot, Naive Bayes được so sánh với mô hình Logistic Regression, thực hiện trên tập huấn luyện có kích thước chỉ với 55 cặp (câu hỏi, intent). Theo đó, tác giả đánh giá mô hình thông qua các thông số accuracy, precision và recall; kết quả chỉ ra hiệu năng của Logistic Regression tốt hơn Naïve Bayes. Tuy nhiên, nghiên cứu không chỉ ra thời gian huấn luyện của mỗi mô hình và chỉ so sánh trên 1 tập

kiểm thử chỉ gồm 11 câu hỏi. Ở trong [20], nhóm nghiên cứu đã ứng dụng lí thuyết Naïve Bayes để phân loại kết quả phỏng vấn cho một chatbot phỏng vấn; nhóm tác giả đã đánh giá bot bằng hình thức khảo sát 50 người tham gia. Theo [14], tác giả đưa ra sự so sánh giữa các mô hình phân loại văn bản; trong đó nói Naïve Bayes cho thời gian huấn luyện nhanh, còn SVM có tốc độ huấn luyện chậm hơn. Trong [21], nhóm nghiên cứu đã sử dụng thư viện scikit-learn, trong đó có Naïve Bayes và SVM để đánh giá các mô hình; kết quả chỉ ra Support Vector Machine có kết quả tốt nhất theo thông số đánh giá accuracy, precision, recall. Đã có nghiên cứu sử dụng mô hình SVM để phân loại văn bản và vận dụng xây dựng chatbot tư vấn khách hàng sử dụng dịch vụ Vietnam Airlines qua Facebook [22]. Theo đó, bài báo đề xuất sử dụng mô hình SVM và đánh giá mô hình đó bằng cách so sánh với các mô hình Naïve Bayes, KNN, DT; nhưng bài báo như chưa đề cập thời gian huấn luyện trên từng mô hình cũng như đánh giá hiệu năng trên các tập kiểm thử có kích thước khác nhau.

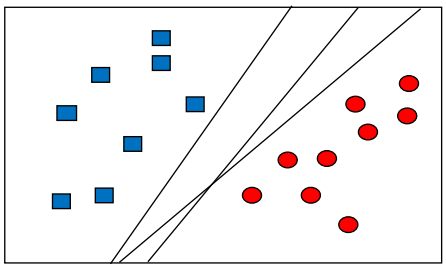
Trong bài báo này, nhóm tác giả so sánh các phương pháp phân loại văn bản Support Vector Machine, Linear Regression và Naïve Bayes. Mục tiêu của nghiên cứu là: (1) So sánh và đánh giá độ chính xác, thời gian huấn luyện của 3 mô hình phân loại văn bản trên các tập kiểm thử (test set) có kích thước khác nhau; (2) Vận dụng kết quả từ (1) đề xuất giải pháp xây dựng một chatbot tư vấn tuyển sinh hoạt động trên nền tảng của Facebook Messenger.

2. Tổng quan về các mô hình phân loại văn bản

Trong mục này, nhóm tác giả giới thiệu sơ lược về khái niệm về các mô hình bài toán cũng như các phương pháp sử dụng để giải quyết các bài toán, bao gồm: Support Vector Machine, Linear Regression sử dụng phương pháp SGD, Multinomial Naïve Bayes.

2.1. Support Vector Machine (SVM)

SVM là một trong những thuật toán phân lớp phổ biến và hiệu quả. Ý tưởng đứng sau thuật toán SVM liên quan đến lý thuyết hình học về khoảng cách các điểm đến một siêu mặt phẳng trong không gian đa chiều.

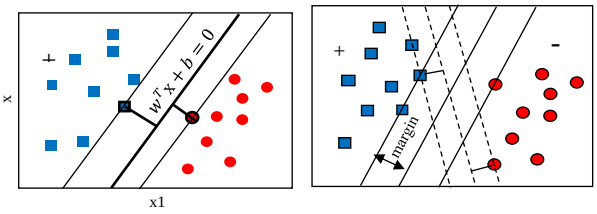


Hình 1. *Có vô số đường thẳng phân tách hai lớp dữ liệu*

Hai lớp dữ liệu ở Hình 1 được giả thiết là có thể tách rời tuyến tính (linearly separable). Khi đó, khoảng cách từ một điểm có tọa độ $(x_{10}, x_{20}, \dots x_{d0})$ tới siêu mặt phẳng d chiều có phương trình $w_1x_1 + w_2x_2 + \cdots + w_dx_d + b = 0$ được xác định bởi:

$$\frac{|w_1x_{10} + w_2x_{20} + \cdots + w_dx_{d0} + b|}{\sqrt{{w_1}^2 + {w_2}^2 + \cdots + {w_d}^2}} = \frac{|w^Tx_0 + b|}{||w||_2} \quad (1)$$

với $x_0 = [x_{10}, x_{20}, \dots x_{d0}]^T$, $w = [w_1, w_2, \dots w_d]^T$.



(a) *(b)*

Hình 2. *(a) Một mặt phẳng phân chia 2 lớp dữ liệu; (b) biên độ (Margin) 2 lớp phải bằng nhau và lớn nhất*

Từ Hình 2a và 2b có thể thấy, biên độ rộng hơn cho kết quả phân lớp tốt hơn vì sự phân chia là rõ ràng hơn. Bài toán tối ưu trong SVM là đi tìm đường phân chia sao cho biên độ giữa 2 lớp là lớn nhất.

2.2. Linear Regression với phương pháp SGD

Giả sử có một tập huấn luyện chứa n cặp (x_i, y_i) với i = 1, 2, …, n; nhiệm vụ của bài toán là tìm giá trị $\hat{y}$ ứng với một vector đầu vào **x** mới. Để làm được điều này, cần phải tìm được mối quan hệ $y \approx f(\mathbf{x})$, với y là giá trị thực của đầu ra dựa trên dữ liệu huấn luyện; $\hat{y}$ là giá trị mà mô hình Linear Regression dự đoán được. Hàm tương quan $y \approx f(\mathbf{x})$ có thể biểu diễn bởi: $y \approx w\bar{x}$, trong đó w là vector hệ số (dạng cột) cần phải tối ưu và $\bar{x}$ là vector đầu vào (dạng hàng). Bài toàn Linear Regression chính là bài toán đi tìm các hệ số tối ưu của vector hệ số w, với mong muốn sai khác giữa y và $\hat{y}$ là nhỏ nhất có thể.

Việc tìm nghiệm của bài toán tối ưu là giải phương trình đạo hàm (gradient) = 0. Một trong những phương pháp để giải là sử dụng thuật toán Stochastic Gradient Descent (SGD) - một biến thể của thuật toán Gradient Descent. Thuật toán Gradient Descent như sau:

- Dự đoán một điểm khởi tạo $\theta = \theta_0$.

- Cập nhật θ đến khi đạt được kết quả chấp nhận được: $\theta = \theta - \eta\nabla_\theta J(\theta)$, với $\nabla_\theta J(\theta)$ là đạo hàm của hàm mất mát tại θ.

Trong bài toán Linear Regression thì $w = \theta$. Với SGD, tại 1 thời điểm chỉ tính đạo hàm của hàm mất mát dựa trên chỉ một điểm dữ liệu x_i rồi cập nhật θ dựa trên đạo hàm này; được tiến hành với từng điểm trên toàn bộ dữ liệu.

2.3. Multinomial Naive Bayes

Bài toán ở đây là đi xác định lớp c của mỗi điểm dữ liệu bằng cách chọn ra lớp mà điểm đó có xác suất rơi vào cao nhất. Công thức Bayes chỉ ra:

$$p(c|X) = \frac{p(X|c)p(c)}{p(X)} \quad (2)$$

Trong đó:

$X = (x_1, x_2, \dots, x_d)$ là vector đầu vào, d là số từ trong từ điển;

p(c|X): xác suất để đầu vào X rơi vào lớp c;

p(c): xác suất một điểm bất kỳ thuộc vào lớp c;

p(X|c): phân phối các điểm dữ liệu trong lớp c.

Giả thiết Naïve Bayes nói rằng tất cả các từ (x_i) trong câu X đều tồn tại độc lập và bình đẳng với nhau. Nghĩa là tất cả các từ đều có tầm ảnh hưởng như nhau đến kết quả đầu ra. Thành phần p(X|c) được tính toán với giả thiết

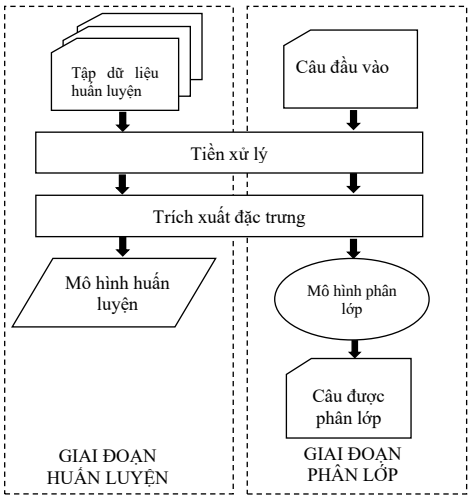
naïve Bayes như sau:

$$p(X|c) = p(x_1, x_2, \dots, x_d \mid c) = \prod_{i=1}^{d} p(x_i \mid c) \quad (3)$$

Trong nghiên cứu này, nhóm tác giả sử dụng mô hình Multinomial Naïve Bayes (MNB). MNB chủ yếu được sử dụng trong phân loại văn bản khi vector đặc trưng được xây dựng dựa trên ý tưởng Bag of Words (BoW) [23]. Nó giúp cải thiện bài toán phân loại văn bản khi mà tập dữ liệu không cân đối giữa các lớp [24].

3. Xây dựng và đánh giá các mô hình huấn luyện

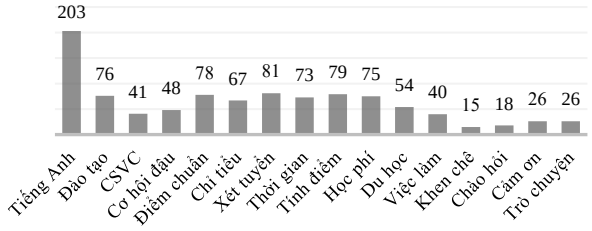
Trong mục này, nhóm tác giả sẽ trình bày các bước xây dựng mô hình phân lớp văn bản để huấn luyện cho chatbot. Để huấn luyện, dữ liệu đầu vào phải trải qua giai đoạn tiền xử lý và trích xuất đặc trưng. Cấu trúc của hệ thống phân lớp được trình bày ở Hình 3.



Hình 3. *Mô hình của hệ thống phân lớp câu hỏi [25]*

3.1. Xây dựng tập dữ liệu

Tập dữ liệu (data set) được xây dựng theo định dạng D = {(x(1), y(1)),…, (x(n), y(n))}, x(i) là các câu hội thoại, y(i) là lớp tương ứng của x(i) nằm trong một tập hữu hạn *K* các lớp được định nghĩa trước. Chatbot cần được huấn luyện trên bộ dữ liệu này với một mô hình có chức năng phân lớp một câu hội thoại mới vào một trong các intent thuộc tập *K*. Trong nghiên cứu này, tập dữ liệu có D = 1000 cặp (câu hỏi, intent); số các lớp K = 16, ứng với 16 chủ đề tư vấn tuyển sinh khác nhau.



Hình 4. *Các lớp dữ liệu và số lượng câu hỏi trong mỗi lớp*

Hình 4 mô tả tập dữ liệu gồm 1000 câu hỏi tư vấn tuyển sinh; bao quát 16 chủ đề với số lượng câu hỏi mỗi chủ đề khác nhau (được đánh số thứ tự từ 0 đến 15 ở các Bảng 3, 4, 5). Bên cạnh các chủ đề phổ biến, nhóm tác giả xây dựng thêm các chủ đề: 'Khen chê', 'Chào hỏi', 'Cảm ơn', 'Trò chuyện' để giúp chatbot hiểu được các lời chào, cảm ơn… nhằm mạng lại cảm giác trò chuyện tự nhiên nhất.

3.2. Tiền xử lý ngôn ngữ

Giai đoạn tiền xử lý ngôn ngữ có nhiệm vụ loại bỏ các khoảng trống, các từ ngữ đệm, những từ không có ý nghĩa khi tham gia vào phân loại văn bản. Trong nghiên cứu này, chức năng word_tokenize trong bộ thư viện nltk được sử dụng nhằm tách từ (word segment).

3.3. Trích xuất đặc trưng và vector hóa dữ liệu

Để có thể sử dụng các giải thuật trong các bài toán máy học, dữ liệu văn bản đầu vào phải được vector hoá. Ở đây phương pháp Bag-of-Words (BoW: túi từ) và TF-IDF được sử dụng để trích xuất đặc trưng. Các thư viện được sử dụng lấy từ thư viện nltk [26]. Phương pháp BoW dựa vào số lượng các từ trong từng loại để xây dựng vector đặc trưng cho từng câu văn bản đầu vào.

3.4. Áp dụng các mô hình phân loại

Như đã giới thiệu ở Mục 1, trong nghiên cứu này tác giả sử dụng các mô hình Support Vector Machine, Naïve Bayes, Linear Regression được lấy từ scikit-learn [27]. Chi tiết các thư viện được trình bày ở Bảng 1.

Bảng 1. *Thư viện scikit-learn sử dụng cho mỗi mô hình*

Mô hình	Phương pháp	Thư viện sklearn
Naïve Bayes	Multinomial Naïve Bayes	sklearn.naive bayes, MultinomialNB
SVM	Support Vector Classifier	sklearn.svm, SVC
Linear Regression	Stochastic Gradient Descent	sklearn.linear model, SGDClassifier

3.5. So sánh đánh giá các mô hình

Nếu phân chia tập dữ liệu thành tập huấn luyện và tập kiểm thử theo tỉ lệ 1: 9 thì có tập huấn luyện gồm 100 cặp (câu hỏi, intent), tập kiểm thử (test set) gồm 900 câu hỏi. Thay đổi tỉ lệ phân chia này sẽ được các tập kiểm thử có kích thước khác nhau. Trong quá trình thực thi, nhóm tác giả thực hiện bằng cách thay đổi thông số test-size (thông số kích thước tập kiểm thử).

3.5.1. So sánh độ chính xác trên các test set khác nhau

Độ chính xác (accuracy) là tỉ lệ giữa số điểm được phân loại đúng trên tổng số điểm trong tập kiểm thử. Phương pháp đánh giá dựa vào thông số độ chính xác được sử dụng nhiều nhất cho các mô hình phân loại văn bản [28].

Bảng 2. *Thông số độ chính xác của các mô hình với các tập kiểm thử có số lượng câu hỏi kiểm thử khác nhau*

Tập kiểm thử	Độ chính xác		
	Naïve Bayes	Support Vector Machine	Linear Regression
100	0,9600	0,9600	0,9600
200	0,9550	0,9650	0,9750
300	0,9500	0,9600	0,9700
400	0,9765	0,9625	0,9600
500	0,9500	0,9580	0,9460
600	0,9080	0,9380	0,9300
700	0,8914	0,9214	0,8814
800	0,8700	0,8988	0,8375
900	0,8100	0,8611	0,6455

Bảng 2 trình bày kết quả về mức độ chính xác trên các tập kiểm thử có kích thước khác nhau. Dựa vào đó có thể thấy, khi tập kiểm thử nhỏ thì độ chính xác cho kết quả tốt và tương đối đồng đều. Khi kích thước tập kiểm thử lớn hơn, bắt đầu có sự chênh lệch đáng kể về mức độ chính xác. Đơn cử trường hợp test set 900: phương pháp SGD của Linear Regression chỉ cho độ chính xác 0,6455; MNB là 0,81; riêng SVM vẫn cho kết quả tốt là 0,8611.

3.5.2. So sánh các mô hình với tập kiểm thử gồm 900 câu hỏi

Với các bài toán có nhiều lớp mà kích thước dữ liệu của các lớp không đồng đều, chỉ một thông số accuracy là chưa đủ để đánh giá hiệu năng của một mô hình; khi này cần dựa vào cặp thông số: precision – recall [29]. Precision được hiểu là có bao nhiêu dự đoán "positive" là thật sự "true" trong thực tế. Recall được hiểu là có bao nhiêu dự đoán "positive" là đúng do mô hình đưa ra. Đôi khi cặp thông số precision và recall là chưa đủ để đánh giá một mô hình, mà còn phải dựa vào F1: giá trị trung bình hài hòa được tính dựa vào precision và recall.

Bảng 3. *Kết quả kiểm thử với mô hình Naive Bayes*

Lớp chủ đề	precision	recall	f1-score
0	0,691	0,989	0,813
1	0,985	0,985	0,985
2	0,783	0,486	0,600
3	0,818	0,837	0,828
4	0,869	0,757	0,809
5	0,879	0,967	0,921
6	0,800	0,767	0,783
7	0,757	0,803	0,779
8	0,955	0,901	0,928
9	0,945	0,765	0,846
10	0,804	0,837	0,820
11	0,784	0,806	0,795
12	1,000	0,429	0,600
13	1,000	0,062	0,118
14	1,000	0,609	0,757
15	1,000	0,217	0,357

Bảng 4. *Kết quả kiểm thử với mô hình SVM*

Lớp chủ đề	precision	recall	f1-score
0	0,721	0,989	0,834
1	0,986	1,000	0,993
2	0,640	0,432	0,516
3	0,949	0,860	0,902
4	0,971	0,943	0,957
5	0,937	0,983	0,959
6	0,899	0,849	0,873
7	0,849	0,939	0,892
8	0,985	0,901	0,941
9	0,847	0,897	0,871
10	0,979	0,939	0,958
11	0,882	0,833	0,857
12	1,000	0,357	0,526
13	1,000	0,062	0,118
14	0,714	0,522	0,686
15	0,897	0,217	0,333

Bảng 5. *Kết quả kiểm thử với Linear Regression*

Lớp chủ đề	precision	recall	f1-score
0	0,377	1,000	0,548
1	1,000	0,750	0,857
2	1,000	0,027	0,053
3	0,909	0,698	0,789
4	1,000	0,643	0,783
5	0,964	0,900	0,931
6	0,887	0,753	0,815
7	0,947	0,818	0,878
8	0,978	0,634	0,769
9	0,979	0,676	0,800
10	1,000	0,204	0,339
11	1,000	0,167	0,286
12	0,000	0,000	0,000
13	0,000	0,000	0,000
14	0,000	0,000	0,000
15	1,000	0,043	0,083

Nhận xét chung các Bảng 3, 4, 5: SVM vẫn có kết quả tốt và ổn định nhất, NB có kết quả tốt tương đối tốt. Đối với những lớp như 12, 13 thì NB và SVM đều có precision tối đa, recall thấp; trong khi đó Linear Regression đều cho precision = recall = 0. Hai lớp 12, 13 này tương ứng 2 lớp 'Khen chê', 'Chào hỏi', có kích thước (số câu hỏi) thấp nhất trong số 16 lớp của tập dữ liệu.

Để dễ dàng đối sánh hiệu năng của 3 mô hình, nhóm tác giả trình bày thông số macro-average. Macro-average được sử dụng để đánh giá khi có sự chênh lệch về số lượng dữ liệu ở mỗi lớp [29]. Ở đây, macro-average precision, macro-average recall và macro F1-score được hiểu lần lượt là trung bình cộng của các precision, recall, F1 score của các lớp dữ liệu.

Bảng 6. *Thông số macro-average precision, recall, f1-score của các mô hình ở test set 900 câu hỏi*

macro-avg	precision	recall	f1-score
Naïve Bayes	0,879	0,701	0,734
SVM	0,897	0,733	0,764
Linear Regression	0,753	0,457	0,496

Nhận xét, mô hình Linear Regression sử dụng SGD cho các thông số macro-avg thấp nhất với f1-score chưa vượt 50%. Còn SVM vẫn nổi trội hơn cả; trong khi đó Naïve Bayes vẫn thể hiện hiệu năng tốt.

3.5.3. So sánh thời gian huấn luyện

Nhằm so sánh thời gian huấn luyện của từng mô hình với thông số kích thước tập huấn luyện được thay đổi từ 0,1 đến 0,9, nhóm tác giả sử dụng hàm thời gian trong Python để tính thời gian từ lúc bắt đầu đến lúc hoàn thành quá trình huấn luyện, kết quả như Bảng 7.

Bảng 7. *Thời gian huấn luyện của các mô hình*

Thông số kích thước tập huấn luyện	Thời gian huấn luyện (ms)		
	Naïve Bayes	Support Vector Machine	Linear Regression
0,1	37,7	35	41
0,2	63	85,1	58,7
0,3	80,1	106	86,3
0,4	88,2	137	124
0,5	115	133	166
0,6	116	155	208
0,7	119	305	237
0,8	155	226	266
0,9	213	205	301

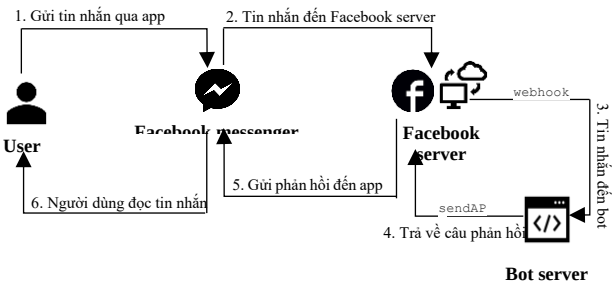
Từ Bảng 7, ở kích thước tập huấn luyện 0,1 và 0,9: SVM cho thời gian nhanh nhất. Với kích thước tập huấn luyện 0,2 thì Linear Regression cho kết quả huấn luyện nhanh nhất. Đối với các training size còn lại thì Naïve Bayes đều trội hơn 2 mô hình kia.

4. Xây dựng chatbot tư vấn tuyển sinh

Từ những kết quả nghiên cứu ở Mục 3 có thể thấy, mô hình Naïve Bayes có độ chính xác tốt và thời gian huấn luyện nhanh. Trong mục này, tác giả trình bày mô hình một chatbot trên nền tảng NodeJs của Facebook Messenger với mô hình Multinomial Naïve Bayes. Công việc này gồm 2 phần chính: (1) Cấu hình ứng dụng bằng tài khoản Facebook developer, và (2) xây dựng phần back-end cho chatbot. Phần (1) khá là đơn giản bởi đã có rất nhiều tài liệu phổ biến sẵn. Vì thế trong phần này tác giả tập trung mô tả phần (2): cấu hình, xây dựng phần back-end của chatbot bằng ngôn ngữ Python, chạy trên nền tảng NodeJs; mã nguồn của bot được deploy ở server của Heroku.

4.1. Mô hình chatbot đề xuất

Tin nhắn người dùng gửi đi thông qua app Messenger sẽ được gửi đến tài khoản nhà phát triển ứng dụng tại Facebook server. Tin nhắn gửi đến bot server thông qua cơ chế webhook và bot phản hồi về bằng sendAPI của Facebook. Cuối cùng câu phản hồi sẽ được gửi đến người dùng thông qua app Messenger. Thứ tự các bước được trình bày ở Hình 5.



Hình 5. *Mô hình hoạt động chatbot Facebook Messenger*

4.2. Cấu hình chatbot

4.2.1. Các khởi tạo cần thiết

Trước hết, để phát triển một ứng dụng trên nền tảng NodeJs, nhà phát triển cần khai báo các thông số như express, body-parser, request, … phục vụ cho việc đẩy dữ liệu đã nhập lên HTML. Bên cạnh đó, để phục vụ cho quá trình tiền xử lý ngôn ngữ và trích xuất, nhóm tác giả khai báo các thư viện stemmer, node-vntokenizer [30].

4.2.2. Cấu hình webhook

Webhook (có thể gọi là web callback) cho phép các ứng dụng cung cấp dữ liệu trong thời gian thực.

Trước hết, Facebook cần xác minh rằng webhook đang hoạt động. Nhà phát triển ứng dụng cần tạo một mã thông báo tuỳ chọn để đăng ký webhook nhận các sự kiện cho bot. Khi đó, Facebook gửi một request bằng mã thông báo đã được cung cấp trong tham số hub.verify. Webhook sẽ xác minh mã thông báo chính xác và gửi phản hồi trở lại qua thông số hub.challenge. Nếu mã không chính xác sẽ có phản hồi 403. Khi nhận được sự kiện webhook, bot luôn phải trả về phản hồi 200. Messenger sẽ gửi lại sự kiện webhook sau mỗi 20 giây cho đến khi nhận được phản hồi. Nếu bot không phản hồi sẽ bị messenger hủy đăng ký.

4.2.3. Xây dựng tập dữ liệu huấn luyện

Sử dụng 1000 cặp (câu hỏi, intent) đã được xây dựng và trình bày ở Phần 3.1. Mỗi cặp như vậy được thêm vào tập dữ liệu huấn luyện theo function training data.push.

4.2.4. Xây dựng mô hình huấn luyện

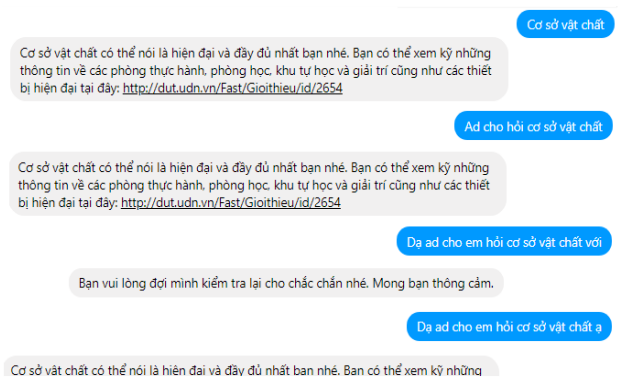
Hai thư viện ‘stemmer’ và ‘vntokenizer’ được sử dụng lần lượt cho giai đoạn tiền xử lý và trích xuất đặc trưng. Riêng ở bước huấn luyện, thay vì sử dụng thư viện của scikit-learn, chúng tôi vận dụng mô hình phân phối xác suất Multinomial Naïve Bayes để phát triển thuật toán của mô hình phân lớp.

Phương pháp thực hiện với Multinomial Naïve Bayes như sau: Một câu hội thoại được đưa bởi người dùng sẽ đi qua các bước tiền xử lý nhằm loại bỏ các ‘stopwords’ và trích xuất đặc trưng bởi phương pháp BoW; Sau đó mô hình phân lớp sẽ xác định ‘score’ của câu hỏi đó cho từng intent trong tập dữ liệu. Kết quả intent đầu ra chính là intent có ‘score’ cao nhất.

5. Kết quả và đánh giá thực nghiệm

5.1. Kết quả thực nghiệm

Kết quả thu được qua triển khai thực nghiệm khá khả quan. Hình 6, minh họa một trường hợp người dùng hỏi 4 câu liên quan về cơ sở vật chất. Chatbot đã trả lời đúng 3 trên 4 câu. Khi ‘score’ tính được bằng 0, chatbot được lập trình để đưa câu trả lời “Bạn vui lòng đợi để mình kiểm tra lại cho chắc chắn nhé. Mong bạn thông cảm”.



Hình 6. *Một kết quả thực nghiệm trên lớp ‘Cơ sở vật chất’*

5.2. Đánh giá thực nghiệm

Nhóm tác giả thực hiện quá trình đánh giá thực nghiệm trên 50 người dùng - những người được yêu cầu hỏi chatbot những câu liên quan đến vấn đề tuyển sinh. Nội dung chủ đề (intent) cần hỏi và số lượng câu hỏi mỗi chủ đề được nhóm tác giả cung cấp cho mỗi người. Tập kiểm thử (tổng số câu hỏi mà mỗi người đưa ra trên mọi chủ đề) lúc này là 200 câu. Đối chiếu lại với Hình 4 ở Mục 3.1, số câu ở mỗi chủ đề mà mỗi người phải hỏi chatbot bằng 20% (tỉ lệ 200:1000) số câu hỏi trong mỗi chủ đề của tập dữ liệu. Số liệu cụ thể được trình bày ở cột (1) và (2) ở Bảng 8.

Tính toán thông số độ chính xác của chatbot bằng cách lấy tỉ lệ % giữa câu trả lời đúng thu được so với số câu hỏi đặt ra. Ở cột (3), tính toán số câu trả lời đúng trung bình cho mỗi lớp dữ liệu bằng cách lấy giá trị trung bình cộng của số câu trả lời đúng thu được từ 50 người. Sau cùng, số lượng câu trả lời đúng trung bình trên toàn bộ các lớp tính được là 155.1 câu. Tổng số câu hỏi là 200, từ đó suy ra độ chính xác của chatbot là 0,775.

Bảng 8. *Độ chính xác trung bình của các lớp dữ liệu*

Intent	Số câu huấn luyện	Số câu hỏi kiểm thử	Số câu trả lời đúng trung bình	Độ chính xác
(0)	**(1)**	**(2)**	**(3)**	**(4)**
0	203	40	29,4	0,74
1	76	15	10,9	0,73
2	41	8	6,40	0,80
3	48	10	6,80	0,68
4	78	16	14,2	0,89
5	67	13	9,30	0,72
6	81	16	13,0	0,82
7	73	15	11,0	0,74
8	79	16	13,0	0,82
9	75	15	11,8	0,79
10	54	11	9,10	0,83
11	40	8	7,00	0,88
12	15	3	2,00	0,67
13	18	4	3,20	0,80
14	26	5	4,10	0,82
15	26	5	3,90	0,78
Tổng cộng			**155,1**	**0,775**

Dựa vào số liệu Bảng 8 có thể thấy, chatbot trả lời một số chủ đề rất hiệu quả như lớp 4 (Điểm chuẩn) với độ chính xác 0,89, hoặc là 0,88 đối với lớp 11 (Việc làm). Nhưng chatbot cũng trả lời với độ chính xác thấp (0,68) ở lớp 3 (Cơ hội đậu). Các lớp như 0, 1, 5, 7, 12 cho độ chính xác xấp xỉ 0,7. Tóm lại, nếu so sánh độ chính xác thực nghiệm là 0,775 với độ chính xác 0,955 của Multinomial Naïve Bayes, tập kiểm thử 200 ở Bảng 2, thì kết quả thu được 0,775 là chưa tốt như mong đợi. Giả sử người dùng hỏi 10 câu thì chatbot đưa ra 7 đến 8 câu trả lời đúng. Mặc dù kết quả thực nghiệm còn phụ thuộc vào nhiều yếu tố khách quan và chủ quan, tuy nhiên kết quả thu được vẫn rất đáng quan tâm.

6. Kết luận

Trong bài báo này, nhóm tác giả đã so sánh các mô hình phân loại văn bản dựa trên các thuật toán học máy SVM, Linear Regression, và Naïve Bayes. Kết quả đã cho thấy mô hình sử dụng SVM có độ chính xác tốt nhất, tuy nhiên thời gian huấn luyện lâu. Trong khi đó, Naïve Bayes cho kết quả độ chính xác tốt thứ 2 và có thời gian huấn luyện nhanh nhất. Từ đó, nhóm tác giả đã sử dụng thuật toán Naïve Bayes để xây dựng chatbot tư vấn tuyển sinh qua Facebook, và cho kết quả thực nghiệm tốt, có thể ứng dụng tại các đơn vị giáo dục.

Lời cảm ơn: Bài báo này được tài trợ bởi Trường Đại học Bách khoa – Đại học Đà Nẵng với đề tài có mã số T2019-02-54.

TÀI LIỆU THAM KHẢO

[1] Hakan Sundblad, Question Classification in Question Answering Systems, Submitted to Linköping Institute of Technology at Linköping University in partial fulfilment of the requirements for the degree of Licentiate of Philosophy, Linköping 2007.

[2] H. N. Io and C. B. Lee, "Chatbots and conversational agents: A bibliometric analysis," 2017 *IEEE International Conference on Industrial Engineering and Engineering Management* (IEEM), Singapore, 2017, pp. 215-219.

[3] Rainer Winkler, Matthias Sollner, Unleashing the Potential of Chatbots in Education: *A State-Of-The-Art Analysis, 78th annual meeting of the academy of management*, Chicago, Illinois, 3/2018.

[4] Ho Thao Hien, Pham-Nguyen Cuong, Le Nguyen Hoai Nam, Ho Le Thi Kim Nhung and Le Dinh Thang. 2018. Intelligent Assistants in HigherEducation Environments: *The FIT-EBot, a Chatbot for Administrative and Learning Support*. In SoICT' 18: *Ninth International Symposium on Information and Communication Technology*, Da Nang City, Viet Nam. ACM, New York, NY, USA, 8 pages, December 6–7, 2018.

[5] Hussain S., Ameri Sianaki O., Ababneh N, A Survey on Conversational Agents/Chatbots Classification and Design Techniques. In: *Barolli L., Takizawa M., Xhafa F., Enokido T. (eds) Web, Artificial Intelligence and Network Applications*. WAINA 2019. Advances in Intelligent Systems and Computing, vol 927. Springer, Cham.

[6] Jagdish Singh, Minnu Helen Joesph and Khurshid Begum Abdul Jabbar, Rule-based chabot for student enquiries, Journal of Physics: *Conference Series*, et al 2019 J. Phys.: Conf. Ser. 1228 012060, 2019

[7] Daniel Jurafsky & James H. Martin Dialog, *Systems and Chatbots, in Speech and Language Processing*, Chapter 24, 2018.

[8] Yang and Xin Liu, "A re-examination of text categorization methods", *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'99), 1999.

[9] Joachims, "Text Categorization with Support Vector Machines, Learning with Many Relevant Features", *European Conference on Machine Learning (ECML)*, 1998.

[10] Marina Sokolova, Guy Lapalme, "A systematic analysis of performance measures for classification tasks", *Information Processing & Management*, Volume 45, Issue 4, p427-437, 07/2009.

[11] Bùi Khánh Linh, Nguyễn Thị Thu Hà, Nguyễn Thị Ngọc Tú, Đào Thanh Tĩnh, "Phân Loại Văn Bản Tiếng Việt Dựa Trên Mô Hình Chủ Đề", *Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX ―Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR'9),* Cần Thơ, ngày 4-5/8/2016, 2016.

[12] Andrew Mccallum, Kamal Nigam, *A Comparison of Event Models for Naive Bayes Text Classification*, 5/2001.

[13] L Douglas Baker, Andrew Kachites McCallum, *Distributional clustering of words for text classification*, 1998.

[14] A Survey Report on Text Classification with Different Term Weighing Methods and Comparison between Classification

Algorithms, *International Journal of Computer Applications* (0975 – 8887) Volume 75– No.7, August 2013.

[15] Vũ Hữu Tiệp, "*Machine Learning Cơ Bản*", Chương 11: Naive Bayes Classifier, trang 128, 8/2018.

[16] Diab Shadi, "Optimizing Stochastic Gradient Descent in Text Classification Based on Fine-Tuning Hyper-Parameters Approach. A Case Study on Automatic Classification of Global Terrorist Attacks", *International Journal of Computer Science and Information Security (IJCSIS)*, 18/02/2019.

[17] "Application of Doc2vec and Stochastic Gradient Descent algorithms for Text Categorization", *Journal of Information Hiding and Multimedia Signal Processing*, Volume 9, Number 5, 09/2018.

[18] Jupudi, Lakshmi, "Stochastic Gradient Descent using Linear Regression with Python", *International Journal of Advanced Research and Applications, (IJA-ERA)*, Volume 2, Issue 8, 12/01/2016.

[19] Helmi Setyawan, Muhammad Yusril, Awangga Rolly Maulana, Efendi Safif, "Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot", 01/10/2018.

[20] Moechammad Sarosa, Mochammad Junus, Mariana Ulfah Hoesny, Zamah Sari, Martin Fatnuriyah, Classification Technique of Interviewer-Bot Result using Naïve Bayes and Phrase Reinforcement Algorithms, *International Journal of Emerging Technologies in Learning*, Vol 13, No 02, 2018.

[21] Chaitrali S. Kulkarni, Amruta U. Bhavsar, Savita R. Pingale, Prof. Satish S. Kumbhar., Bank chatbot – An Intelligent Assistant System Using NLP and Machine Learning, *International Research Journal of Engineering and Technology (IRJET)*, Volume: 04 Issue: 05, May -2017.

[22] Nguyễn Thành Thủy, *Ứng dụng thuật toán học có giám sát multi-class SVM trong xây dựng hệ thống chatbot hỏi đáp tiếng Việt*, The 7th conference on information technology and its applications, 2018

[23] Vũ Hữu Tiệp, Chương 11: Naïve Bayes Classifier, trang 129, *Machine Learning Cơ Bản*, 01/3/2018.

[24] E. Frank, and R. R. Bouckaert, Naive bayes for text classification with unbalanced classes, Knowledge Discovery in Databases: PKDD, pp 503-510, 2006.

[25] Vũ Thị Tuyến, Một số mô hình học máy trong phân loại câu hỏi. Luận văn thạc sĩ CNTT. Trường Đại học Công Nghệ, Đại học Quốc gia Hà Nội, 2016.

[26] Natural Language Toolkit, [online] https://www.nltk.org/

[27] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[28] Marina Sokolova, Guy Lapalme, "A systematic analysis of performance measures for classification tasks", *Information Processing & Management*, Volume 45, Issue 4, p427-437, 07/2009.

[29] Vũ Hữu Tiệp, bài 33: Các phương pháp đánh giá một hệ thống phân lớp, *Machine Learning Cơ Bản*, 01/3/2018.

[30] Duyetdev, https://github.com/duyetdev/node-vntokenizer, latest commit Aug 17, 2019.

(BBT nhận bài: 29/11/2019, hoàn tất thủ tục phản biện: 03/7/2020)