Applying Multi-Class SVMs into building chatbot Vietnamese question answering system

Conference Paper · September 2018

CITATIONS
0

READS
1,841

1 author:

Thuy Nguyen-Thanh
The University of Danang
8 PUBLICATIONS **10** CITATIONS

SEE PROFILE

ỨNG DỤNG THUẬT TOÁN HỌC CÓ GIÁM SÁT MULTI-CLASS SVM TRONG XÂY DỰNG HỆ THỐNG CHATBOT HỎI ĐÁP TIẾNG VIỆT

Nguyễn Thành Thủy[1]

[1]Trường Đại học Kinh tế, Đại học Đà Nẵng
thuynt@due.edu.vn

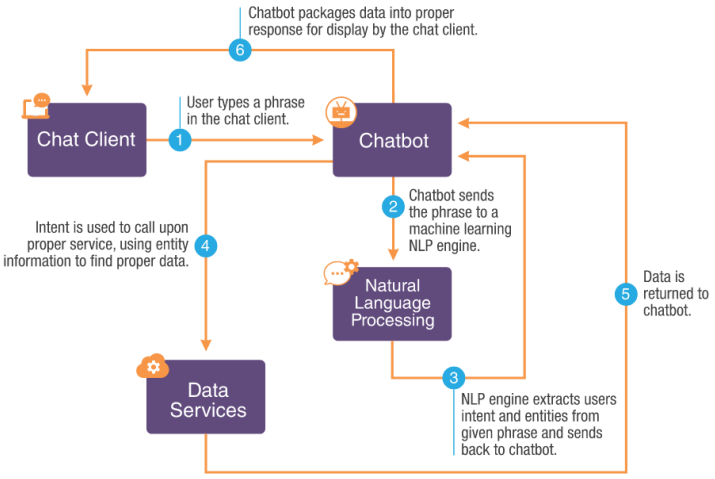
Abstract. Việc xác định ý định của người dùng đóng vai trò quan trọng trong thiết kế hệ thống chatbot, nó sẽ quyết định đến câu trả lời hay hành vi kế tiếp của bot. Trong nghiên cứu này, chúng tôi đề xuất một giải pháp ứng dụng thuật toán học có giám sát Multi-Class SVM (Support Vector Machine) để xây dựng hệ thống chatbot hỏi – đáp tiếng Việt, mô hình học máy sẽ giúp bot hiểu và giao tiếp được với con người thông qua đàm thoại văn bản. Trong đó, chúng tôi sử dụng kỹ thuật túi từ BoW (Bag of Words) kết hợp với phương pháp túi từ TF-IDF (Term Frequency – Inverse Document Frequency) để xây dựng vector đặc trưng ngữ nghĩa của các câu văn bản tiếng Việt, sử dụng thuật toán Multi-Class SVM để huấn luyện và tiến hành phân lớp, so sánh độ chính xác với các thuật toán khác. Bot hiểu được ý định người dùng thông qua độ tương đồng ngữ nghĩa giữa câu hỏi đầu vào với tập không gian câu hỏi – câu trả lời được sử dụng trong bước huấn luyện. Cuối cùng, chúng tôi đã ứng dụng giải pháp trên để cài đặt mô phỏng hệ thống chatbot, hỗ trợ trả lời tự động các câu hỏi thường gặp của khách hàng khi sử dụng dịch vụ của Vietnam Airlines.

Keywords: Chatbot, Multi-class SVM, BoW, TF-IDF.

1 Giới thiệu

Chatbot (Trợ lý ảo) là một chương trình máy tính tương tác với người dùng bằng ngôn ngữ tự nhiên dưới một giao diện đơn giản, thông qua âm thanh (giọng nói) hoặc văn bản. Chatbot là một hình thức thô sơ của phần mềm trí tuệ nhân tạo, hoạt động độc lập, có thể tự động trả lời những câu hỏi hoặc xử lý tình huống càng thật càng tốt [1]. Độ phức tạp của bài toán tập trung vào câu hỏi là làm sao Bot có thể hiểu được ý định (Intents) của con người thông qua một câu hỏi đầu vào. Sau khi hiểu được ý định của con người thì hệ thống dễ dàng tương tác và đề xuất câu trả lời phù hợp nhất.

Có hai mô hình Chatbot chính, (1) Mô hình ứng dụng trong miền đóng (closed domain), trả lời theo mô hình truy xuất thông tin (retrieval-based model). Trong đó, Bot đưa ra câu trả lời đã được chuẩn bị trước hoặc tuân theo những mô thức nhất định, thường sử dụng trong các hoạt động hỗ trợ chăm sóc khách hàng hoặc trợ lý mua sắm trực tuyến. (2) Mô hình ứng dụng trong miền mở (open domain), người dùng có thể thực hiện cuộc trò chuyện với bot ở mọi nơi, không có một mục tiêu hay ý định rõ ràng, không giới hạn chủ đề [4].



Hình 1. Cơ chế hoạt động chung của một ChatBot.

Nghiên cứu về hệ thống hỏi đáp tự động (Question Answering - QA) hiện đang thu hút sự quan tâm của các nhà nghiên cứu, có ý nghĩa khoa học lẫn ý nghĩa thực tế. Nhiều hội nghị thường niên về khai phá dữ liệu, trích chọn thông tin dành một chủ đề riêng cho các nghiên cứu về hệ thống hỏi đáp như TREC [9], CLEF [10],… Ngoài ra còn có các phần mềm thương mại liên quan đến QA cũng được phát triển như Yahoo Answers, Google QnA, Live QnA, Answers.com của Answer Corp, Ask của InterActive Corp, M của Facebook,…

Hiện nay đã có một số nghiên cứu về bài toán phân lớp câu hỏi, đặc biệt là tiếng Anh như nghiên cứu của Zhiheng Huang và các cộng sự [16]. Nghiên cứu của Dell Zhang và Wee Sun Lee [2],… Hầu hết các thực nghiệm đều cho thấy kết quả phân lớp sử dụng thuật toán SVM đạt được độ chính xác cao nhất.

Trong nghiên cứu này, chúng tôi đề xuất ứng dụng phương pháp học có giám sát Multi-Class SVM phân lớp câu hỏi trong miền đóng, hỗ trợ xây dựng mô phỏng chatbot hỏi-đáp. Chúng tôi đã sử dụng mô hình túi từ BoW kết hợp với phương pháp xác định trọng số của từ TF-IDF để xây dựng vector đặc trưng ngữ nghĩa của các câu hỏi, sử dụng thuật toán Multi-Class SVM để huấn luyện và tiến hành phân lớp. Sau đó, ứng dụng phương pháp này để xây dựng thực nghiệm hệ thống chatbot hỗ trợ trả lời tự động các câu hỏi thường gặp của khách hàng khi sử dụng dịch vụ của hãng Hàng Không VietNam Airlines.

2 Bài toán phân lớp ý định người dùng (Intents)

2.1 Phát biểu bài toán

Đối với miền ứng dụng đóng, chúng ta có thể giới hạn rằng số lượng Intent nằm trong một tập hữu hạn những Intent đã được định nghĩa sẵn. Với giới hạn này, bài toán xác định ý định người dùng có thể quy về bài toán phân lớp văn bản. Với đầu vào là một câu giao tiếp của người dùng, hệ thống phân lớp sẽ xác định Intent tương ứng trong tập các Intent đã được định nghĩa [4].

Håkan Sundblad [5] đã đưa ra một định nghĩa phân lớp câu hỏi như sau: "*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 và* $C=\{c_1, c_2, ..., c_{|c|}\}$ *là tập các phân lớp cho trước.*" Cặp (q_j,c_i) được gán cho giá trị là T chỉ ra rằng câu hỏi q_j thuộc phân lớp c_i và được gán cho giá trị là F nếu q_j không thuộc phân lớp c_i.

Bài toán phân lớp câu hỏi có thể được phát biểu như sau:

Input:

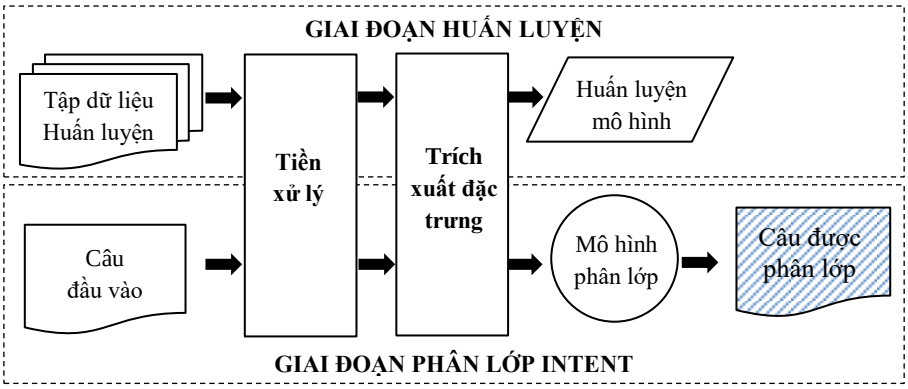
- Cho trước một tập các câu hỏi: $Q = \{q_1, q_2,\ldots, q_n\}$
- Tập các lớp được định nghĩa: $C = \{c_1, c_2,\ldots, c_n\}$

Output:

- Nhãn c_i của câu hỏi q_j.

2.2 Giải bài toán theo phương pháp học máy

Các kỹ thuật học máy (Machine Learning) sẽ thay thế các kiến thức chuyên môn bằng một tập lớn các câu hỏi được gán nhãn (tập dữ liệu huấn luyện), sử dụng tập này, mô hình phân lớp sẽ được huấn luyện có giám sát. Một số thuật toán thường được sử dụng như: Mạng nơ-ron, Naïve Bayes, Maximum Entropy, Decision Tree, Nearest-Neighbors, SNoW, SVM,... Cách tiếp cận bằng học máy đã giải quyết được các hạn chế trong cách tiếp cận dựa trên luật, đây là cách tiếp cận được sử dụng phổ biến để giải quyết bài toán phân lớp câu hỏi.



Hình 2. Kiến trúc của hệ thống phân lớp Intent [13]

Phân lớp câu hỏi theo kỹ thuật học có giám sát bao gồm 2 giai đoạn chính: **giai đoạn huấn luyện** và **giai đoạn phân lớp**. (Hình 2)

Bài toán phân lớp câu hỏi cho hệ thống chatbot mà chúng tôi đang hướng đến, được xây dựng trong miền dữ liệu đóng. Dữ liệu đầu vào là tập các cặp (Câu hỏi, Câu trả lời) độc lập đã được gán nhãn Intent (ý định), các Intent ở đây chính là mục tiêu của người hỏi được gắn với câu trả lời cụ thể.

3 Thiết kế mô hình huấn luyện học máy

3.1 Xây dựng tập dữ liệu huấn luyện

Để xây dựng một mô hình phân lớp Intent, chúng ta cần một tập dữ liệu huấn luyện bao gồm các cách diễn đạt khác nhau cho mỗi Intent. Ví dụ, người dùng có thể diễn đạt theo nhiều cách khác nhau với cùng một mục đích hỏi, như sau:

- Giống chó nào không được vận chuyển trên chuyến bay dưới dạng hàng hóa?
- Giống chó nào tôi không gửi được theo đường hàng hóa trên chuyến bay?
- Hàng hóa của tôi có chứa động vật như là chó có được không?
- Tôi muốn gửi chó qua đường hàng hóa trên máy bay có được không?
- Tôi có thể đưa chó lên máy bay theo đường hàng hóa hay không?
- …

Nguồn dữ liệu thực nghiệm, chúng tôi đã thu thập từ 35 bộ (Câu hỏi, Câu trả lời), là những câu hỏi thường gặp của khách hàng khi sử dụng dịch vụ của Vietnam Airlines [14], bộ dữ liệu được tổ chức trên tập **D={(q_1,a_1), (q_2,a_2), … (q_{35},a_{35})}.**

Tập D được tách thành hai tập con tập: Câu_hỏi và Câu_trả_lời, được lưu trong hai file: **Questions.csv** và **Answers.csv**, theo cú pháp: **<Intent>|<Question/Answer**>.

Để làm giàu cho tập dữ liệu huấn luyện, chúng tôi đã tiến hành bổ sung thêm 19 câu hỏi mới trong mỗi Intent, mỗi câu hỏi là một các cách diễn đạt khác nhau nhưng có cùng mục đích với câu hỏi (ban đầu) trong tập D (*để cho khách quan, người tham gia xây dựng bộ câu hỏi đến từ nhiều vùng miền, độ tuổi khác nhau*). Trong đó, mỗi câu hỏi trong tập D chính là một Intent trong tập dữ liệu huấn liệu, tập nhãn: Intent={1, 2,.., 35}.

Tập dữ liệu huấn luyện thu được gồm 700 cặp (Question, Intent), **T={($q_{1.k}$,k), ($q_{2.k}$,k),…, ($q_{20.k}$,k)} (k=[1..35]).** Được tổ chức trong file **Questions_Extend.scv**, theo cú pháp: **<Intent>|<Question>.**

3.2 Tiền xử lý văn bản tiếng Việt

- *Làm sạch dữ liệu văn bản*: chuẩn hóa chữ tiếng Việt không dấu sang có dấu, chuẩn hóa "i" và "y", lỗi sai chính tả, chuẩn hóa font, dấu câu, xóa các từ dừng (stopwords),...
- *Tách mỗi câu thành một danh sách các từ tố (token):* Mỗi câu được tách ra thành một danh sách các từ có nghĩa.
- *Chuẩn hóa từ đồng nghĩa*: đồng nhất từ đồng nghĩa, từ địa phương, tiếng lóng về một từ chuẩn hóa.
- *Xác định từ loại (part of speech: từ loại):* Sau khi câu được tách thành danh sách các từ. Bước này sẽ xác định đúng từ loại (POS - như noun, verb, pronoun, adverb ...) của mỗi từ trong câu.

Ví dụ, xâu: "*Hàng hóa của tôi có chứa động vật như là chó có được không?*"
Sau tiền xử lý: "*Hàng_hóa chứa động_vật chó*".

Sau khi tiền xử lý, văn bản có thể xem như là một tập hợp các đặc trưng, đó là tập hợp các từ quan trọng còn lại để biểu diễn văn bản. Việc phân loại văn bản sẽ dựa trên các đặc trưng này.

Trong khâu tiền xử lý, chúng tôi đã sử dụng các thư viện mở để cài đặt: từ điển stopwords của Van-Duyet Le [12]; thư viện ViTokenizer, ViPosTagger của Viet-Trung Tran [15]. Ngoài ra, để rút ngắn số chiều không gian đặc trưng, mô hình BoW kết hợp với thuật toán TF.IDF (được trình bày ở mục 3.3) có thể giúp loại bỏ những từ lặp lại nhiều lần (những từ không quan trọng) trong văn bản.

Sau khi tiền xử lý, tập T gồm 700 cặp (Question, Intent), được lưu vào tập tin **Questions_Extend700.scv** để đưa vào xây dựng vector đặc trưng ở bước tiếp theo.

3.3 Trích xuất đặc trưng và vector hóa văn bản

Để số hóa văn bản, chúng tôi đã sử dụng mô hình Bag-of-Words (BoW) để xây dựng vector đặc trưng, kết hợp thuật toán TF-IDF để xác định giá trị các phần tử trong vector.

Mô hình BoW là một mô hình được sử dụng phổ biến trong lĩnh vực phân loại văn bản. Trong đó, mỗi văn bản d_i trong tập ngữ liệu đang xét, tập này có n câu văn bản và m từ xuất hiện không lặp lại (theo từng cặp), sẽ được mô hình hóa như là một vector trọng số của các đặc trưng: $\vec{d}_i = \{w_{i1}, w_{i2}, \ldots, w_{im}\}$, w_{ij} là trọng số của đặc trưng thứ j ($1 <= j <= m$).

Để xác định trọng số (sự quan trọng) của các từ trong văn bản, hiện nay thuật toán đánh giá từ khóa dựa trên sự kết hợp của độ đo cục bộ và toàn cục là TF-IDF cho một kết quả khá tốt. Độ đo trọng số của một từ w_{ij} trong tài liệu d_i sẽ được tính bằng công thức sau:

$$w_{ij} = TF_{ij} * \log\left(\frac{N}{DF_j}\right) [12]$$

- TF_{ij} là số lần xuất hiện của từ thứ j trong văn bản d_i
- DF_j là tổng số văn bản có chứa từ thứ j trong tập ngữ liệu
- N là tổng số văn bản trong tập ngữ liệu

Ví dụ, cho tập ngữ liệu gồm 2 câu văn bản sau tiền xử lý như sau:

d_1 = "*chó vận_chuyển chuyến_bay hàng_hóa*"

d_2 = "*gửi hành_lý vận_chuyển đường hàng_hóa*"

Thực hiện vector hóa văn bản bằng mô hình BoW, kết hợp với thuật toán TF-IDF để xác định trọng số, cho kết quả như sau:

$\overrightarrow{d_1}$ = {0.08, 0.08, 0.00, 0.00, 0.00, 0.00, 0.00}

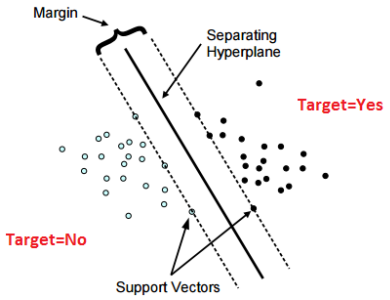
$\overrightarrow{d_2}$ = {0.00, 0.00, 0.06, 0.06, 0.00, 0.06, 0.00}

Tập T chúng tôi đã xây dựng sau tiền xử lý, gồm 700 câu hỏi (văn bản) và từ điển có n từ vựng (không lặp lại theo từng cặp). Tập T được biểu diễn thành ma trận có kích thước 700 x n, dòng thứ i của ma trận T là vector trọng số (đặc trưng) của câu hỏi thứ i, để đưa vào huấn luyện mô hình ở bước tiếp theo.

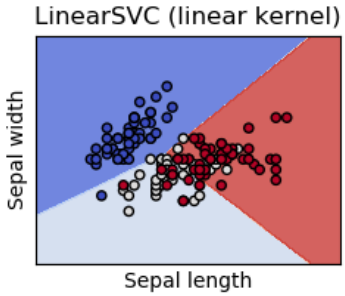
4 Thuật toán học có giám sát Multi-Class SVM

Giải thuật máy vector hỗ trợ SVM (Support Vector Machine) ra đời từ lý thuyết học thống kê do Vapnik và Chervonenkis xây dựng năm 1995 [6]. Đây là một giải thuật phân lớp có hiệu quả cao và đã được áp dụng nhiều trong lĩnh vực khai phá dữ liệu và nhận dạng. Ban đầu thuật toán SVM được thiết kế để giải quyết bài toán phân lớp nhị phân (Hình 3). Để phân nhiều lớp (Multi-Class) thì kỹ thuật SVM nguyên

thủy sẽ chia không gian dữ liệu thành 2 phần và quá trình này lặp lại nhiều lần. Giả sử bài toán cần phân loại có k lớp (k > 2), chiến lược "một-đối-một" sẽ tiến hành $\frac{k(k-1)}{2}$ lần phân lớp nhị phân, mỗi lớp sẽ tiến hành phân tách với k-1 lớp còn lại để xác định k-1 hàm phân tách dựa vào bài toán phân hai lớp bằng SVM (Hình 4) [6].



Hình 3. 2-Class SVM



Hình 4. Multi-Class SVM

Áp dụng Multi-Class SVM vào bài toán phân loại câu hỏi

Tập huấn luyện T đã được biểu diễn trong không gian vector đặc trưng, gồm có 35 Intents tương ứng với 35 lớp cần huấn luyện. Trong đó mỗi tài liệu là một điểm, phương pháp này giúp tìm ra các siêu phẳng quyết định tốt nhất có thể chia các điểm trên không gian này thành 35 lớp riêng biệt.

5 Kết quả thực nghiệm

5.1 Thực nghiệm phân lớp câu hỏi

Để đánh giá tính hiệu quả của phương pháp được đề xuất trong xây dựng hệ thống chatbot hỏi đáp tiếng việt, chúng tôi đã tiến hành cài đặt giải thuật Multi-Class SVM trên Python, có sử dụng bộ thư viện scikit-learn [8]. Thực hiện so sánh hiệu quả của mô hình Multi-Class SVM với các mô hình khác, bao gồm: Naive Bayes (NBs), k-Nearest Neighbors (kNN) và Decision Tree (DT). Tất cả thực nghiệm được thực hiện trên môi trường PC (Intel Xeon(R), CPU E3-1230 V2 @3.30GHz, 3.70 Ghz, 8GB RAM), hệ điều hành Windows 10.

Tập dữ liệu gồm 700 câu hỏi, được chia làm 35 Intent (tương ứng 35 bộ câu hỏi – câu trả lời gốc [14]), mỗi Intent gồm 20 câu hỏi có cùng mục đích. Các bước tổ chức dữ liệu, tiền xử lý và trích xuất đặc trưng đã được nêu ở trên.

Với tập dữ liệu nhỏ này, để tránh hiện tượng overfitting và underfitting khi xây dựng mô hình, chúng tôi đã sử dụng kỹ thuật Leave-One-Out (một trường hợp của k-Fold cross validation) để tổ chức tập Training-set và Test-set trong quá trình huấn luyện và đánh giá mô hình. Nhằm tăng hiệu quả cho mỗi mô hình khi cài đặt, chúng tôi sử dụng phương pháp GridSearch để tối ưu hóa các tham số (parameter). Kết quả tối ưu tham số cho các mô hình được trình bày trong Bảng 1.

Để đánh giá mô hình cho bài toán phân lớp nhiều lớp, chúng tôi sử dụng các chỉ số: Accuracy, Macro-average Precision, Macro-average Recall và Macro-average F1-Score để so sánh, được thể hiện trong Bảng 2.

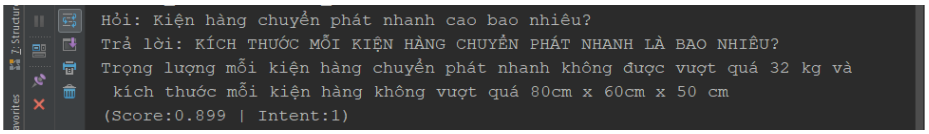
Bảng 1. Tối ưu hóa các tham số

Model	Parameters
SVM	kernel='sigmoid', C=5000.00, gamma=0.0005, class_weight='balanced'
NBs	alpha=0.10
kNN	n_neighbors= 1.00
DT	max_depth=77.00

Bảng 2. So sánh hiệu quả phân loại câu hỏi giữa các mô hình thực nghiệm

Các chỉ số	SVM	NBs	kNN	DT
Accuracy	**0.87429**	0.81714	0.76857	0.63857
Macro-average Precision	0.78435	0.78052	0.68011	0.56584
Macro-average Recall	0.77427	0.77430	0.67108	0.54732
Macro-average F1-Score	0.77928	0.77740	0.67557	0.55643

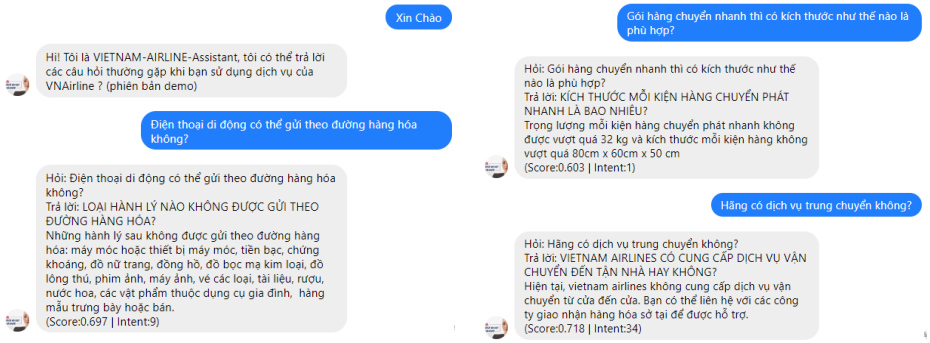
Bảng 2, cho ta thấy giải thuật phân lớp SVM có độ chính xác Accuracy và F1-Score nhỉnh hơn giải thuật NBs (0.05 - 0.001) và trội hơn nhiều so với giải thuật kNN (0.11 - 0.103) và DT (0.24 - 0.222).



Hình 5. Một mẫu phân lớp trên hệ thống khi chạy thực nghiệm

5.2 Thực nghiệm hệ thống chatbot

Để cài đặt hệ thống chatbot, chúng tôi đã sử dụng framework DialogFlow, sử dụng PC local tại phòng lab thực hiện code python để làm server xử lý payload (webhook server), server được kết nối với DialogFlow bằng công cụ Dynamic DNS No-IP giúp bot có thể hoạt động liên tục.



Hình 6. Thực nghiệm chatbot VIETNAM-AIRLINES-Assistant trên facebook messenger

Chatbot VIETNAM-AIRLINES-Assistant, sử dụng các kỹ thuật phân lớp đã nêu trên, được cài đặt thử nghiệm thành công trên Facebook, Skype, Slack, Zalo và nền web (*thực nghiệm tại đây* [3]) kết quả hoạt động khá ổn định. Để tăng tính hiệu quả khi xác định Intent của người dùng, chúng tôi đã kết hợp cả 3 thuật toán SVM, NBs và kNN để xác định Intent phù hợp nhất với ngữ cảnh dựa vào predicted score trả về ở mỗi mô hình. Mã nguồn được chúng tôi phổ biến tại đây [11].

6 Kết luận và hướng phát triển

Trong bài viết này chúng tôi đã trình bày phương pháp phân loại câu hỏi tiếng Việt trong miền dữ liệu đóng, dựa trên thuật toán học có giám sát Multi-Class SVM và ứng dụng mô hình học máy này để xây dựng ứng dụng chatbot hỏi-đáp. Kết quả thực nghiệm mô hình với tập dữ liệu thực cho thấy phương pháp của chúng tôi đề xuất là khá hiệu quả khi so sánh với các giải thuật NBs, kNN và DT, độ chính xác đạt đến 87.5%. Hệ thống chatbot thực nghiệm hoạt động có hiệu suất như kỳ vọng.

Để nâng cao tính hiệu quả của mô hình trên, chúng tôi cần bổ sung thêm số câu hỏi được gán nhãn trên mỗi Intent trong tập dữ liệu huấn luyện. Trong tương lai, chúng tôi tiếp tục nghiên cứu mô hình học bán giám sám (Semi-Supervised Learning), để bot tự học dựa trên những câu hỏi mới được đưa vào của người dùng trong quá trình vận hành hệ thống chatbot.

Danh mục tài liệu tham khảo

1. ChatBots ORG Homapge. Truy xuất https://www.chatbots.org/.
2. Dell Zhang and Wee Sun Lee (2003). Question classification using support vector machines. *In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR '03)*. ACM, New York, NY, USA, pages 26-32.
3. Demo Chatbot VIETNAM-AIRLINES-Assistant. Truy xuất https://www.hoctructuyen123.net/p/chatbot-demo.html
4. FPT Tech Insight Homepage. 3 vấn đề NLP cơ bản khi phát triển một hệ thống chatbot và một số phương pháp giải quyết điển hình. Truy xuất https://tech.fpt.com.vn/3-van-de-nlp-co-ban-khi-phat-trien-mot-thong-chatbot-va-mot-phuong-phap-giai-quyet-dien-hinh.
5. Håkan Sundblad (2007). Question Classification in Question Answering Systems. *Linköping*.
6. Nguyễn Đức Vinh (2009). Phân tích câu hỏi trong hệ thống hỏi đáp Tiếng Việt. *Khóa luận tốt nghiệp đại học*. Trường Đại học Công Nghệ, Đại học Quốc gia Hà Nội.
7. Stefan Kojouharov. "*Ultimate Guide to Leveraging NLP & Machine Learning for your Chatbot*". Truy xuất https://chatbotslife.com/ultimate-guide-to-leveraging-nlp-machine-learning-for-you-chatbot-531ff2dd870c#.rabx346bq.
8. Supervised Learning, scikit-learn. Truy xuất http://scikit-learn.org/stable/supervised_learning.html#supervised-learning
9. Text Retrieval Conference (TREC). Truy xuất https://trec.nist.gov/.
10. The Cross-Language Evaluation Forum (CLEF). http://clef.isti.cnr.it/
11. Thủy Nguyễn Thành. *Mã nguồn dự án Chatbot.* Truy xuất https://github.com/thuy-nguyenthanh/VIETNAM-AIRLINES-Assistant-PROJECT/.
12. Van-Duyet Le. Truy xuất https://github.com/stopwords/vietnamese-stopwords/.
13. Vũ Thị Tuyến (2016). 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.
14. Vietnam Airlines. *Những câu hỏi thường gặp*. Truy xuất https://www.vietnamairlines.com/vi/cargo/customer-support/faqs/.
15. Viet-Trung Tran. Truy xuất https://pypi.org/project/pyvi/.
16. Zhiheng Huang, Marcus Thint and Zengchang Qin. Question Classification using Head Words and their Hypernyms. *Proceedings of the 2008 Conference on Empirical Methods in atural Languae Processing*. Pages 927-936, Honolulu, October 2008.