Tạp chí Khoa học Trường Đại học Cần Thơ
Phần A: Khoa học Tự nhiên, Công nghệ và Môi trường
website: sj.ctu.edu.vn

DOI:10.22144/ctu.jvn.2019.093

GIẢI PHÁP PHÂN LOẠI BÀI BÁO KHOA HỌC BẰNG KỸ THUẬT MÁY HỌC

Trần Thanh Điện[1*], Thái Nhựt Thanh[2] và Nguyễn Thái Nghe[3]

[1]*Nhà xuất bản Đại học Cần Thơ, Trường Đại học Cần Thơ*
[2]*Tạp chí khoa học Trường Đại học Cần Thơ*
[3]*Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ*
**Người chịu trách nhiệm về bài viết: Trần Thanh Điện (email: thanhdien@ctu.edu.vn)*

Thông tin chung:
Ngày nhận bài: 08/07/2019
Ngày nhận bài sửa: 09/08/2019
Ngày duyệt đăng: 29/08/2019

Title:
An approach to scientific paper classification using machine learning

Từ khóa:
Bayes thơ ngây, k-láng giềng gần nhất, máy học véc-tơ hỗ trợ, phân loại văn bản

Keywords:
k Nearest Neighbor, Naïve Bayes, Support Vector Machine, Text classification

ABSTRACT

Publication of research is the ultimate and significant step to recognize scientific work. However, in the submission system with a wide range of subjects (e.g. Association for Computing Machinery with 2,000 subjects), it may take the authors a lot of time to classify a manuscript into an appropriate group of subjects before it is submitted to a journal or conference. Therefore, this article is aimed to propose automatic solutions to extract information and categorize scientific papers on suitable topics. The experiments was based on the data set of scientific articles published in Can Tho University Journal of Science. The input data were pre-processed, extracted, vectorized and classified using three machine learning techniques including support vector machines, Naïve Bayes, and k-nearest neighbors. The experimental results showed that with the accuracy of over 91%, support vector machines technique proved its feasibility for developing the automatic classification system of scientific papers.

TÓM TẮT

Nghiên cứu khoa học là một phần không thể thiếu trong các trường đại học, viện nghiên cứu, phòng thí nghiệm và cả các công ty lớn. Kết quả của các công trình nghiên cứu khoa học thường được trình bày dưới dạng các bài báo được gửi đến các tạp chí, hội thảo. Tuy nhiên, các hệ thống nhận bài của các tạp chí lớn có rất nhiều chủ đề như Hiệp hội quốc tế về nghiên cứu, giáo dục ngành khoa học máy tính ACM có hơn 2.000 chủ đề, do vậy các tác giả và ban biên tập mất khá nhiều thời gian khi xác định một bài viết thuộc nhóm chủ đề nào trước khi nộp bài cho các tạp chí, hội thảo. Bài viết này đề xuất giải pháp tự động rút trích thông tin và phân loại một bài báo khoa học vào chủ đề nào đó. Dữ liệu vào sẽ được tiền xử lý, rút trích, véc-tơ hóa và phân loại bằng kỹ thuật máy học. Thực nghiệm được xây dựng trên tập dữ liệu là các bài báo khoa học đã được gửi đăng trên Tạp chí khoa học của Trường Đại học Cần Thơ. Các kỹ thuật máy học véc-tơ hỗ trợ (SVM), Bayes thơ ngây (Naïve Bayes), và k-láng giềng gần nhất (kNN) đã được sử dụng để so sánh nhằm tìm ra kết quả tốt nhất. Kết quả thực nghiệm cho thấy kỹ thuật SVM đã cho độ chính xác > 91%, rất khả thi cho việc xây dựng hệ thống tự động phân loại bài báo khoa học.

Trích dẫn: Trần Thanh Điện, Thái Nhựt Thanh và Nguyễn Thái Nghe, 2019. Giải pháp phân loại bài báo khoa học bằng kĩ thuật máy học. Tạp chí Khoa học Trường Đại học Cần Thơ. 55(4A): 29-37.

1 GIỚI THIỆU

Với sự phát triển bùng nổ của thông tin và sự phát triển đồng thời của khả năng tính toán tự động thì phân loại dữ liệu, đặc biệt là dữ liệu văn bản có tầm đặc biệt quan trọng (Thaoroijam, 2014). Phân loại là một kỹ thuật học có giám sát (supervised learning), được ứng dụng nhiều trong thực tế như định tuyến trung tâm cuộc gọi (call center routing), trích xuất siêu dữ liệu tự động (automatic metadata extraction) (Li *et al.*, 2017). Trong lĩnh vực máy học (machine learning) và xử lý ngôn ngữ tự nhiên (natural language processing - NLP), phân loại văn bản (text classification) là một bài toán xử lý văn bản cổ điển, nhằm phân một văn bản mới vào nhóm các văn bản cho trước dựa trên sự tương đồng của văn bản đó so với nhóm văn bản (Sebastiani, 2002). Theo Yang and Liu (1999) thì phân loại văn bản là việc gán các nhãn phân loại lên một văn bản mới dựa trên mức độ tương tự của văn bản đó so với các văn bản đã được gán nhãn trong tập huấn luyện. Phân loại văn bản tự động giúp cho việc lưu trữ, tìm kiếm thông tin nhanh chóng hơn. Ngoài ra, với số lượng văn bản lớn thì thao tác phân loại lần lượt trong từng văn bản sẽ mất rất nhiều thời gian, công sức, chưa kể khả năng xảy ra trường hợp phân loại không chính xác do tính chủ quan của người phân loại. Các ứng dụng phân loại văn bản rất đa dạng như lọc thư rác (spam email), phân loại tin tức theo chủ đề trên các báo điện tử, quản lý tri thức và hỗ trợ cho các công cụ tìm kiếm trên Internet (Thaoroijam, 2014).

Một vấn đề đang được hội đồng biên tập của các tạp chí khoa học quan tâm là làm sao phân loại một bài viết gửi đăng vào một lĩnh vực phù hợp của tạp chí. Chẳng hạn, hệ thống nộp bài tự phân loại lĩnh vực (chủ đề), rút trích các thông tin liên quan một cách tự động khi tác giả gửi (upload) một bài viết lên hệ thống, đặc biệt đối với tạp chí lớn như Hiệp hội quốc tế về nghiên cứu, giáo dục ngành khoa học máy tính (Association for Computing Machinery - ACM) với hơn 2.000 chủ đề thì tác giả mất rất nhiều thời gian để xác định chủ đề của bài viết. Vì vậy, việc tự động xác định chủ đề của một bài viết là rất cần thiết.

Vấn đề phân loại văn bản được nhiều nhà khoa học quan tâm với các hướng tiếp cận khác nhau. Một cách tiếp cận được nhiều nhà nghiên cứu sử dụng là phương pháp máy học, với nhiều thuật toán giải bài toán phân loại văn bản như: k láng giềng gần nhất (k nearest neighbor - kNN), Naïve Bayes, máy học véc-tơ hỗ trợ (support vector machines - SVM), cây quyết định (decision tree), mạng neuron nhân tạo (artificial neural network) (George and Pat, 1995; McCallum and Nigam, 1998; Liu *et al.*, 2003; Chen *et al.*, 2009; Aggarwal and Zhai, 2012; Bijaksana *et al.*, 2013; Zhang *et al.*, 2013; Haddoud *et al.*, 2016).

Nghiên cứu này đề xuất giải pháp phân loại tự động bài báo khoa học nhằm hỗ trợ các tác giả, ban biên tập phân loại lĩnh vực của bài báo khi nộp bài trực tuyến. Bài báo là tập tin dạng .doc(x) hoặc .pdf, khi nộp vào, hệ thống sẽ rút trích thông tin các tác giả, tựa bài, tóm tắt/abstract, đặc biệt là việc xác định lĩnh vực của bài viết (chẳng hạn lĩnh vực: Công nghệ thông tin, Môi trường, Thủy sản…).

2 PHÂN LOẠI VĂN BẢN VÀ CÁC NGHIÊN CỨU LIÊN QUAN

2.1 Phân loại văn bản

Phân loại văn bản tự động là việc phân chia một tập văn bản đầu vào thành hai hoặc nhiều lớp, trong đó mỗi văn bản có thể thuộc một hoặc nhiều lớp. Công việc này nhằm mục đích gán nhãn (hay lớp - class) được định nghĩa trước cho các văn bản. Ví dụ: Gán nhãn cho mỗi bài viết mới trên một báo điện tử vào một trong các chủ đề như Công nghệ, Thể thao, Giải trí; gán nhãn tự động cho mỗi bài viết gửi đăng tạp chí vào một trong các lĩnh vực Công nghệ thông tin, Môi trường, Thủy sản…

Nhiệm vụ phân loại được bắt đầu xây dựng từ một tập các văn bản $D = \{d_1, ..., d_n\}$ được gọi là tập huấn luyện (training set) và trong đó các tài liệu d_i được gán nhãn c_j với c_j thuộc tập các chủ đề $C = \{c_1, ..., c_m\}$. Nhiệm vụ đặt ra là xác định được mô hình phân loại để một tài liệu d_k bất kỳ có thể phân loại chính xác vào một trong những chủ đề của tập chủ đề C. Hay nói cách khác, mục tiêu của bài toán là đi tìm hàm f:

$$f: D \times C \rightarrow Boolean$$

$$f(d, c) = \begin{cases} true, & \text{nếu } d \text{ thuộc lớp } c \\ false, & \text{nếu } d \text{ không thuộc lớp } c \end{cases}$$

2.2 Các giải thuật phân loại văn bản

Có nhiều thuật toán phân loại văn bản. Trong bài viết này, nhóm tác giả sử dụng ba thuật toán kNN, Naïve Bayes, SVM. Đây là ba thuật toán được nhiều nghiên cứu đánh giá là hiệu quả trong phân loại văn bản.

2.2.1 Giải thuật kNN

kNN là giải thuật phân loại (hay phân lớp) các đối tượng dựa vào khoảng cách gần nhất giữa đối tượng cần phân lớp và tất cả các đối tượng trong tập huấn luyện (Tan *et al.*, 2006). Ý tưởng của phương pháp này là khi cần phân loại một văn bản mới, thuật toán sẽ tính toán khoảng cách của tất cả các văn bản trong tập huấn luyện đến văn bản này để tìm ra tập K láng giềng gần nhất.

Để phân lớp cho một văn bản mới *x*, trước hết bộ phân lớp sẽ tính khoảng cách từ văn bản *x* đến tất cả các văn bản trong tập huấn luyện. Qua đó tìm được tập *N(x, D, k)* gồm *k* văn bản mẫu có khoảng cách đến *x* là gần nhất.

Thuật toán kNN được mô tả như sau:

1. Xác định giá trị tham số *k* (số láng giềng gần nhất).

2. Tính khoảng cách giữa đối tượng cần phân lớp (*query point*) với tất cả các đối tượng trong tập huấn luyện training data (thường sử dụng khoảng cách *Euclidean*).

3. Sắp xếp khoảng cách theo thứ tự tăng dần và xác định *k* láng giềng gần nhất với đối tượng cần phân lớp.

4. Lấy tất cả các lớp của *k* láng giềng gần nhất đã xác định.

5. Dựa vào phần lớn lớp của láng giềng gần nhất để xác định lớp cho đối tượng cần phân lớp *query point*.

2.2.2 Giải thuật Naïve Bayes

Thuật toán Naïve Bayes (Mitchell, 1997) là một thuật toán phổ biến trong máy học được McCallum and Nigam (1998) và Yang and Liu (1999) đánh giá là một trong những phương pháp có hiệu năng cao nhất khi thực hiện phân lớp văn bản. Ý tưởng cơ bản của cách tiếp cận này là sử dụng xác suất có điều kiện giữa từ hoặc cụm từ và chủ đề để dự đoán xác suất chủ đề của một tài liệu cần phân loại.

Thuật toán Naïve Bayes dựa trên định lý Bayes được phát biểu như sau:

$$P(Y|X) = \frac{P(XY)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)}$$

Áp dụng trong bài toán phân loại, các dữ kiện gồm có: D: tập dữ liệu huấn luyện đã được vector hóa dưới dạng $\vec{x} = (x_1, x_2, \dots, x_n)$; C_i: phân lớp i, với i = {1, 2, …, m}; các thuộc tính độc lập điều kiện đôi một với nhau. Theo định lý Bayes:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Theo tính chất độc lập điều kiện:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

Trong đó, $P(C_i|X)$ là xác suất thuộc phân lớp *i* khi biết trước mẫu *X*; $P(C_i)$ xác suất là phân lớp *i*; $P(x_k|C_i)$ xác suất thuộc tính thứ *k* có giá trị x_k khi đã biết *X* thuộc phân lớp *i*.

Các bước thực hiện thuật toán Naïve Bayes:

Bước 1: Huấn luyện Naïve Bayes (dựa vào tập dữ liệu), tính $P(C_i)$ và $P(x_k|C_i)$

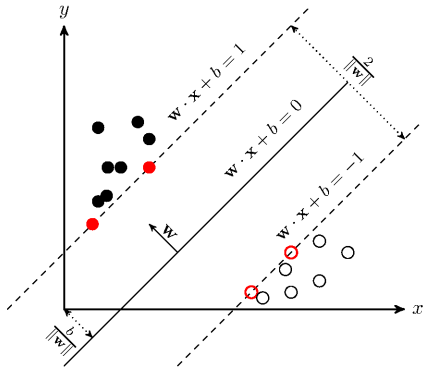
Bước 2: Phân lớp $X^{new} = (x_1, x_2, \dots, x_n)$, ta cần tính xác suất thuộc từng phân lớp khi đã biết trước X^{new}. X^{new} được gán vào lớp có xác suất lớn nhất theo công thức:

$$\max_{C_i \in C} \left(P(C_i) \prod_{k=1}^{n} P(x_k|C_i) \right)$$

2.2.3 Giải thuật thuật SVM

Thuật toán máy SVM được Cortes and Vapnik (1995) giới thiệu lần đầu tiên. SVM rất hiệu quả để giải quyết các bài toán với dữ liệu có số chiều lớn như các véc-tơ biểu diễn văn bản. SVM được xem là bộ phân lớp chính xác nhất cho bài toán phân lớp văn bản (Chakrabarti, 2003) do tốc độ phân lớp rất nhanh và hiệu quả đối với bài toán phân lớp văn bản.

Ý tưởng của phương pháp này là cho trước một tập huấn luyện được biểu diễn trong không gian véc-tơ, trong đó mỗi văn bản được xem là một điểm trong không gian này. Phương pháp này tìm ra một mặt siêu phẳng *h* quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng, gọi là lớp dương (+) và lớp âm (-). Như vậy, bộ phân loại SVM là một mặt siêu phẳng tách các mẫu thuộc lớp dương ra khỏi các mẫu thuộc lớp âm với độ chênh lệch lớn nhất. Độ chênh lệch này hay còn gọi là khoảng cách biên được xác định bằng khoảng cách giữa mẫu (+) và mẫu (-) gần mặt siêu phẳng nhất (Hình 1). Khoảng cách này càng lớn thì các mẫu thuộc hai lớp càng được phân chia rõ ràng, nghĩa là sẽ đạt được kết quả phân loại tốt. Mục tiêu của thuật toán SVM là tìm được khoảng cách biên lớn nhất để tạo được kết quả phân loại tốt.



Hình 1: Siêu phẳng lề cực đại trong không gian hai chiều (Cortes and Vapnik, 1995)

Phương trình mặt siêu phẳng chứa véc-tơ *x* trong không gian đối tượng như sau: $w.x + b = 0$

Trong đó, *w* là véc-tơ trọng số, *b* là độ lệch/thiên vị (bias). Hướng và khoảng cách từ gốc tọa độ đến mặt siêu phẳng thay đổi khi thay đổi *w* và *b*.

Bộ phân lớp SVM được định nghĩa như sau: $f(x) = sign(w.x + b)$

Trong đó: $\begin{cases} f(x) = +1, \ w.x + b \geq 0 \\ f(x) = -1, \ w.x + b < 0 \end{cases}$

Gọi y_i mang giá trị +1 hoặc -1. Nếu y_i = +1 thì *x* thuộc về lớp (+), ngược lại y_i = -1 thì *x* thuộc về lớp (-). Hai mặt siêu phẳng tách các mẫu thành hai phần được mô tả bởi các phương trình: $w.x + b = 1$ và $w.x + b = -1$. Bằng hình học có thể tính khoảng cách giữa hai mặt siêu phẳng này là: $\frac{2}{\|w\|}$

Để khoảng cách biên là lớn nhất cần phải tìm giá trị nhỏ nhất của $||w||$; đồng thời, ngăn chặn các điểm dữ liệu rơi vào vùng bên trong biên, cần thêm ràng buộc sau:

$\begin{cases} w.x_i + b \geq 1, & \text{với mẫu } (+) \\ w.x_i + b \leq -1, & \text{với mẫu } (-) \end{cases}$

Có thể viết lại như sau: $y_i(w.x_i + b) \geq 1, với\ i \in (1,n)$

Khi đó, việc tìm siêu phẳng *h* tương đương giải bài toán tìm *Min*||*w*|| với *w* và *b* thỏa điều kiện sau: $\forall i \in (1,n)$: $y_i(w.x_i + b) \geq 1$

2.2.4 Các thông số đánh giá giải thuật

Khi xây dựng một mô hình máy học, các nhà nghiên cứu cần một phép đánh giá để xem mô hình sử dụng có hiệu quả không và để so sánh khả năng của các mô hình. Trong các bài toán phân lớp thì việc định nghĩa lớp dữ liệu quan trọng hơn cần được xác định đúng là lớp dương (Positive), lớp còn lại được gọi là âm (Negative). Giả sử, để đánh giá một bộ phân loại hai lớp tạm gọi là (+) và (-), khi đó:

– TP (True positive) là số phần tử dương được phân loại dương; FN (False negative) là số phần tử dương được phân loại âm; TN (True negative) là số phần tử âm được phân loại âm; và FP (False positive) là số phần tử âm được phân loại dương.

– Độ chính xác (Precision) được định nghĩa là tỷ lệ số phần tử TP trong số những phần tử được phân loại là positive (TP + FP): $Precision = \frac{TP}{TP+FP}$

– Độ bao phủ (Recall) được định nghĩa là tỷ lệ số phần tử TP trong số những phần tử thực sự là positive (TP + FN): $Recall = \frac{TP}{TP+FN}$

Precision cao đồng nghĩa với việc độ chính xác của các phần tử tìm được là cao. Recall cao đồng nghĩa với việc tỷ lệ bỏ sót các phần tử thực sự positive là thấp.

Độ đo F_1: chỉ số cân bằng giữa độ chính xác và độ bao phủ. Nếu độ chính xác và độ bao phủ cao và cân bằng thì độ đo F_1 lớn, còn độ chính xác và độ bao phủ nhỏ, không cân bằng thì độ đo F_1 nhỏ. Như vậy, F_1 càng cao thì bộ phân lớp càng tốt. Khi cả Recall và Precision đều bằng 1 (tốt nhất có thể) thì F_1 = 1. Khi cả Recall và Precision đều thấp (ví dụ bằng 0.1, tức là F_1 = 0.1) thì phân lớp không tốt.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

2.3 Các nghiên cứu liên quan về phân loại văn bản

Nhiều nghiên cứu phân loại văn bản được ứng dụng để giải quyết các bài toán trong thực tế, có thể ví dụ hai nghiên cứu sau:

2.3.1 Phân loại văn bản bằng SVM và cây quyết định

Nhóm tác giả Trần Cao Đệ và Phạm Nguyên Khang (2012) đã nghiên cứu SVM, áp dụng nó vào bài toán phân loại văn bản và so sánh hiệu quả của nó với hiệu quả của giải thuật phân lớp cổ điển cây quyết định. Ngoài ra, nhóm nghiên cứu đã áp dụng kỹ thuật phân tích giá trị đơn SVD (Singular Value Decomposition) vào giải thuật SVM để rút ngắn số chiều của không gian đặc trưng, làm giảm nhiễu, giúp quá trình phân loại được hiệu quả hơn. Trong giai đoạn tiền xử lý dữ liệu, nhóm tác giả đã sử dụng giải thuật MMSEG (Tsai, 2000) để tiến hành tách từ. Sau khi tách từ tác giả tiến hành mô hình hóa văn bản thành dạng véc-tơ, sử dụng TF*IDF véc-tơ hóa; tiến hành phân loại văn bản với hai giải thuật SVM và cây quyết định trong phần mềm Weka. Với tập dữ liệu là 7842 văn bản thuộc 10 chủ đề khác nhau, ứng với mỗi chủ đề, tác giả chọn ra 500 văn bản một cách ngẫu nhiên để tiến hành huấn luyện, số văn bản còn lại để kiểm chứng độc lập. Kết quả phân loại cho thấy phân lớp với SVM thực sự tốt hơn phân lớp bằng cây quyết định. Ngoài ra, việc dùng SVD để phân tích và rút gọn số chiều của không gian đặc trưng đã giúp nâng cao hiệu quả phân loại với SVM.

2.3.2 Phân loại văn bản với giải thuật Naïve Bayes

Nhóm tác giả Trần Thị Thu Thảo và Vũ Thị Chinh (2012) đã tiến hành xây dựng module tách từ theo mô hình N-gram, sau đó mô hình hóa văn bản đã được tách từ bằng véc-tơ TF*IDF. Với tập dữ liệu đã được mô hình hóa thành véc-tơ, tác giả tiến hành phân loại dựa trên phương pháp Naïve Bayes. Nhóm tác giả xây dựng phần mềm phân loại, tích hợp thêm các chức năng quản lý, sửa, xóa bài báo để tiến hành thử nghiệm trên tập dữ liệu là 281 bài báo khoa học thuộc các chuyên ngành của lĩnh vực công nghệ thông tin. Kết quả phân loại đạt được khá khả quan,

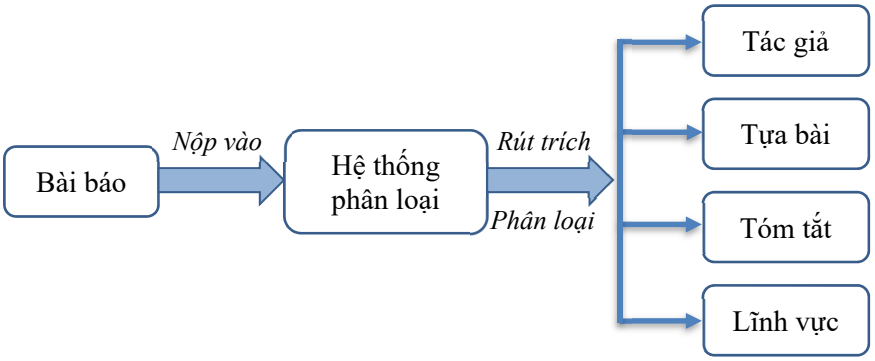
tuy nhiên nghiên cứu còn hạn chế về tập dữ liệu thử nghiệm và chưa có những so sánh đánh giá phương pháp Naïve Bayes với các phương pháp phân loại khác.

3 PHƯƠNG PHÁP NGHIÊN CỨU

3.1 Kiến trúc hệ thống

Kiến trúc tổng thể về hệ thống rút trích thông tin và phân loại lĩnh vực bài báo được mô hình hóa như Hình 2.

Với mô hình này, từ một bài viết mới (dạng .docx, .pdf), tác giả gửi (submit) lên hệ thống quản lý tạp chí, sau đó hệ thống tự động rút trích các thông tin như tên tác giả, tựa bài viết, tóm tắt, đặc biệt phân loại bài viết vào một lĩnh vực phù hợp dựa trên mô hình máy học đã được huấn luyện.



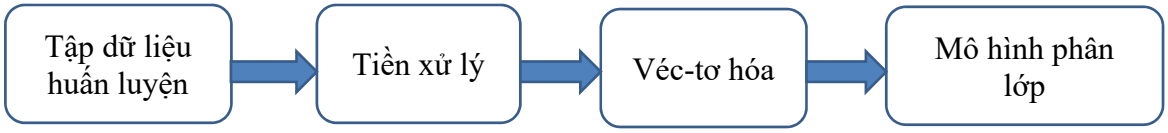
Hình 2: Kiến trúc mô hình rút trích và phân loại bài báo khoa học

Do bài báo là tập tin định dạng sẵn nên việc rút trích thông tin tác giả, tựa bài, tóm tắt là dễ dàng. Vì vậy, nghiên cứu này chỉ tập trung giải quyết vấn đề phân loại lĩnh vực của bài báo khi tác giả gửi đăng tạp chí.

3.2 Các giai đoạn phân loại

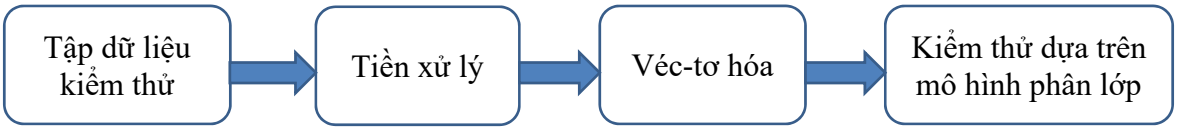
Việc phân loại bài báo khoa học tự động được chia thành hai giai đoạn:

– Giai đoạn huấn luyện: Ở giai đoạn này, dựa vào tập dữ liệu có sẵn đã được phân loại chủ đề cùng với các giải thuật máy học, tiến hành cho máy học để sinh ra mô hình phân lớp (classification model) Giai đoạn này được mô tả như Hình 3.



Hình 3: Giai đoạn huấn luyện

– Giai đoạn kiểm thử: Dựa vào mô hình phân lớp được sinh ra ở giai đoạn huấn luyện để tiến hành phân loại bài báo khoa học ở tập dữ liệu kiểm thử Giai đoạn này được mô tả như Hình 4.



Hình 4: Giai đoạn kiểm thử

3.2.1 Tiền xử lý dữ liệu

Chuyển định dạng .docx sang .txt và chuẩn hóa từ: Do tập dữ liệu được sử dụng là các tập tin định dạng .docx (hoặc .doc) nên cần phải tiến hành chuyển đổi chúng sang định dạng văn bản thuần túy (.txt) để dễ dàng sử dụng trong hầu hết các giải thuật, các thư viện phục vụ cho việc phân loại tự động. Việc chuyển định dạng bài báo đầu vào dựa trên hỗ trợ của Apache POI. Sử dụng Apache POI để thực hiện các thao tác đọc trên tập tin định dạng .docx, sau đó ghi nội dung đọc được vào tập tin .txt. Sau khi chuyển tập tin định dạng từ .docx sang .txt, tiến hành chuẩn hóa từ để chuyển tất cả ký tự của văn bản thành chữ thường, xóa các khoảng trống.

Ví dụ: Câu "Xử Lý Ngôn Ngữ Tự nhiên là 1 nhánh của Trí tuệ nhân tạo" được chuẩn hóa thành "xử lý ngôn ngữ tự nhiên là 1 nhánh của trí tuệ nhân tạo".

Tách từ (word segmention): Trong tiếng Việt, dấu cách (space) không có nhiệm vụ tách từ mà chỉ phân cách giữa các âm tiết. Chính vì vậy, giai đoạn

tách từ cũng khá quan trọng trong xử lý ngôn ngữ tự nhiên.

Hiện tại, có nhiều công cụ được xây dựng thành công để tách từ tiếng Việt với độ chính xác tương đối cao. Nghiên cứu này sử dụng công cụ tách từ *VnTokenizer* được nhóm tác giả Nguyễn Thị Minh Huyền *và ctv.* (2010) phát triển dựa trên cách tiếp cận tổng hợp các phương pháp Maximum Matching, WFST và regular expression parsing, với tập dữ liệu sử dụng là bảng âm tiết tiếng Việt và từ điển từ vựng tiếng Việt. Đây là công cụ tách từ tiếng Việt tự động, tách các văn bản tiếng Việt thành các đơn vị từ vựng (từ ngữ, tên, số, ngày tháng và các biểu thức chính quy khác) với độ chính xác hơn 95%.

Ví dụ: câu "xử lý ngôn ngữ tự nhiên là 1 nhánh của trí tuệ nhân tạo" được tách từ thành "xử_lý ngôn_ngữ tự_nhiên (nlp) là 1 nhánh của trí_tuệ nhân_tạo".

Loại bỏ từ dừng (Stop words): Từ dừng (stop words) là những từ xuất hiện nhiều trong tất cả các văn bản thuộc mọi thể loại trong tập dữ liệu, hay những từ chỉ xuất hiện trong một và một vài văn bản. Nghĩa là stop word là những từ xuất hiện quá nhiều lần và quá ít lần, vì thế nó không có ý nghĩa và không chứa thông tin đáng giá để sử dụng. Trong phân loại văn bản, sự xuất hiện của stop words không những không giúp gì trong việc đánh giá phân loại mà còn nhiễu và giảm độ chính xác của quá trình phân loại (như các từ: thì, là, mà, và, hoặc, bởi…).

Ví dụ: câu "xử_lý ngôn_ngữ tự_nhiên (nlp) là 1 nhánh của trí_tuệ nhân_tạo" khi loại bỏ từ dừng thành "xử_lý ngôn_ngữ tự_nhiên (nlp) nhánh trí_tuệ nhân_tạo".

Trong nghiên cứu, sau khi chuyển bài báo từ định dạng .docx sang .txt và tách từ, nhóm tác giả sử dụng phương pháp loại bỏ từ dừng bằng từ điển từ dừng.

3.2.2 Véc-tơ hóa văn bản

Có một số mô hình biểu diễn văn bản như mô hình không gian véc-tơ (vector space model) dựa trên phương pháp đánh trọng số của từ theo tần số, mô hình túi từ (bag of words model), mô hình hóa văn bản thành đồ thị (graph-based model). Nghiên cứu này đề cập phương pháp biểu diễn văn bản theo mô hình không gian véc-tơ (Perone, 2013). Đây là cách biểu diễn tương đối đơn giản và hiệu quả. Theo mô hình này, mỗi văn bản được biểu diễn thành một véc-tơ; mỗi thành phần của véc-tơ là một từ riêng biệt trong tập văn bản và được gán một giá trị là trọng số của từ đó trong văn bản đó.

Ví dụ: xét 2 văn bản với trọng số là số lần xuất hiện của từ khóa trong văn bản: Văn bản 1 là "Cửa hàng quần áo bán quần áo"; văn bản 2 là "Cửa hàng điện thoại bán điện thoại". Khi đó, mô hình không gian véc-tơ được mô tả như Bảng 1.

Bảng 1: Ví dụ mô hình không gian véc-tơ biểu diễn 2 văn bản

Từ khóa	Văn bản 1	Văn bản 2
Cửa	1	1
Hàng	1	1
Quần	2	0
Áo	2	0
Bán	1	1
Điện	0	2
Thoại	0	2

Bài toán biểu diễn văn bản theo mô hình không gian véc-tơ như sau: Đầu vào là một tập gồm có j văn bản trong miền ứng dụng D, với $D = \{d_1, d_{2,...} d_j\}$ và tập gồm m từ trong mỗi văn bản $T = \{t_1, t_{2,...} t_m\}$; đầu ra lần lượt đánh trọng số cho từng từ trong mỗi văn bản từ đó xây dựng ma trận trọng số w_{ij} là trọng số của từ w_j trong văn bản $d_j \in D$.

Có nhiều giải pháp để đánh trọng số của từ t_i trong văn bản d_j, trong đó giải pháp tích hợp tần số xuất hiện từ khóa (TF - Term Frequency) và nghịch đảo tần số xuất hiện trong các văn bản (IDF- Inverse Document Frequency) được sử dụng khá phổ biến.

TF - Term Frequency: dùng để ước lượng tần suất xuất hiện của một từ trong một văn bản nào đó. Bên cạnh đó, mỗi văn bản đều có độ dài, số lượng từ ngữ khác nhau vì thế số lần xuất hiện của từ sẽ khác nhau. Nên để đánh trọng số của một từ người ta lấy số lần xuất hiện của từ đó chia cho độ dài của văn bản (tức là số từ của văn bản đó).

$$\mathrm{TF}(\mathrm{t_i}, \mathrm{d_j}) = \frac{\text{số lần từ } \mathrm{t_i} \text{ xuất hiện trong văn bản } \mathrm{d_j}}{\text{tổng số từ trong văn bản } \mathrm{d_j}}$$

Các giá trị w_{ij} được tính dựa trên tần số (hay số lần) xuất hiện của từ khóa trong văn bản. Gọi f_{ij} là số lần xuất hiện của từ khóa t_i trong văn bản d_j, khi đó w_{ij} được tính bởi một trong ba công thức cơ bản sau:

$$\begin{cases} w_{ij} = f_{ij} \\ w_{ij} = 1 + \log(f_{ij}) \\ w_{ij} = \sqrt{f_{ij}} \end{cases}$$

Trong đó f_{ij} là số lần xuất hiện của từ khóa t_i trong văn bản d_j.

Nếu t_i xuất hiện trong văn bản d_j thì $w_{ij} = 1$, ngược lại: $w_{ij} = 0$.

IDF - Inverse Document Frequency: dùng để ước lượng mức độ quan trọng của một từ trong một văn bản nào đó. Khi tính tần số TF của một từ thì tất

cả các từ trong tập từ có mức độ quan trọng là như nhau. Tuy nhiên, theo nhiều nghiên cứu cho thấy không hẳn trong một tập dữ liệu tất cả các từ đều quan trọng. Những từ thường không có độ quan trọng cao là: từ nối (nên, nhưng, bên cạnh đó, vì, như vậy…), từ chỉ định (kìa, đó, ấy, thế…), giới từ (trên, trong, ngoài, ở, tại…). Chính những lý do trên mà cần giảm đi mức độ quan trọng của những từ đó bằng cách tính IDF. Công thức tính IDF như sau:

$$IDF(t_i, D) = log \frac{tổng\ số\ văn\ bản\ trong\ tập\ mẫu\ D}{số\ văn\ bản\ có\ chứa\ từ\ t_i}$$

Từ xuất hiện trong nhiều văn bản thì trọng số trong một văn bản sẽ thấp.

Công thức đánh trọng số:

$$\begin{cases} w_{ij} = \log\left(\frac{m}{df_i}\right) = \log(m) - \log(df_i), & nếu\ TF_{ij} \geq 1 \\ w_{ij} = 0, & nếu\ TF_{ij} = 0 \end{cases}$$

TF*IDF: là sự tích hợp giữa tần số xuất hiện từ khóa TF và nghịch đảo tần số xuất hiện trong các văn bản IDF. Phương pháp này khá phổ biến được dùng để tính giá trị TF*IDF của một từ thông qua mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản.

Những từ có IF*IDF cao là những từ xuất hiện nhiều trong văn bản này và xuất hiện ít trong các văn bản khác. Thông qua phương pháp này, có thể lọc ra những từ phổ biến và giữ lại những từ có giá trị cao.

$$TF * IDF(t_i, d_j, D) = TF(t_i, d_j)\ x\ IDF(t_i, D)$$

Công thức đánh trọng số:

$$\begin{cases} w_{ij} = (1 + \log(f_{ij})) \log\left(\frac{N}{df_i}\right), nếu\ f_{ij} \geq 1 \\ w_{ij} = 0 \quad , \text{ nếu } f_{ij} = 0 \end{cases}$$

4 KẾT QUẢ THỰC NGHIỆM

Dữ liệu mẫu sử dụng trong nghiên cứu này là tập dữ liệu gồm 680 bài báo khoa học của 10 lĩnh vực có số bài báo nhiều nhất được xuất bản trên Tạp chí khoa học Trường Đại học Cần Thơ từ năm 2016 đến 2018 (Bảng 2).

Tập dữ liệu bài báo được tiền xử lý bằng cách chuyển định dạng từ tập tin .docx/pdf sang tập tin .txt, sau đó tiến hành tách từ bằng công cụ *vnTokenizer*. Quá trình loại bỏ từ dừng được thực hiện thông qua từ điển từ dừng, số từ còn lại là 4.095 từ. Quá trình mô hình hóa mỗi văn bản là một véc-tơ trọng số các từ. Do đó, tập dữ liệu được mô hình hóa là ma trận chứa TF*IDF của các từ với kích thước 680 * 4.095 phần tử.

Bảng 2: Các bài báo phân bổ trong 10 lĩnh vực của Tạp chí

Stt	Lĩnh vực	Số mẫu huấn luyện	Số mẫu kiểm tra	Tổng số mẫu (bài báo)
1	Công nghệ	45	5	50
2	Môi trường	54	6	60
3	Tự nhiên	54	6	60
4	Chăn nuôi	36	4	40
5	Công nghệ sinh học	27	3	30
6	Nông nghiệp	90	10	100
7	Thủy sản	135	15	150
8	Giáo dục	36	4	40
9	Xã hội - Nhân văn	72	8	80
10	Kinh tế	63	7	70
	Tổng cộng	612	68	680

(Nguồn: Tạp chí khoa học Trường Đại học Cần Thơ)

Sau các quá trình tiền xử lý và véc-tơ hóa, tập dữ liệu các bài báo được huấn luyện với các giải thuật phân loại văn bản tự động như SVM, Naïve Bayes, kNN. Tập dữ liệu các bài báo được phân tách tự động, lấy khoảng 90% dùng làm tập huấn luyện (612 bài báo), 10% còn lại dùng làm tập kiểm tra (68 bài báo). Kết quả phân loại sử dụng 3 thuật toán máy học: SVM, Naïve Bayes, kNN. Việc đánh giá dựa vào các chỉ số độ chính xác (Precision), độ bao phủ (Recall) và độ đo F_1 thể hiện như trong Bảng 3.

Kết quả thực nghiệm cho thấy hiệu quả phân loại của các giải thuật là tương đối tốt. Trong đó, giải thuật SVM có kết quả phân loại tốt nhất, cho độ chính xác > 91%, rất khả thi cho việc xây dựng hệ thống tự động phân loại bài báo khoa học; góp phần giúp cho quá trình phân loại bài báo của tác giả và ban biên tập được nhanh và chính xác hơn. Kết quả phân loại này cũng phù hợp với nhiều nhóm nghiên cứu đã chứng minh bằng thực nghiệm: phương pháp SVM phân loại văn bản cho kết quả tốt tương đương hoặc tốt hơn đáng kể các phương pháp phân loại khác (Boser *et al.*, 1992; Yang and Pedersen, 1997; Burges, 1998; Dumais *et al.*, 1998).

Bảng 3: So sánh kết quả phân loại giữa các giải thuật: SVM, Naïve Bayes, kNN

	SVM			Naïve Bayes			kNN		
Lĩnh vực	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
Công nghệ	0,857	0,857	0,857	0,857	0,857	0,857	0,400	0,571	0,471
Môi trường	1,000	0,333	0,500	0,400	0,333	0,364	0,667	0,333	0,444
Tự nhiên	0,750	1,000	0,857	0,667	0,667	0,667	0,600	1,000	0,750
Chăn nuôi	1,000	1,000	1,000	1,000	0,500	0,667	1,000	0,500	0,667
Công nghệ sinh học	1,000	0,500	0,667	1,000	0,500	0,667	1,000	0,500	0,667
Nông nghiệp	0,786	1,000	0,880	0,846	1,000	0,917	0,733	1,000	0,846
Thủy sản	0,947	1,000	0,973	0,857	1,000	0,923	0,947	1,000	0,973
Giáo dục	1,000	1,000	1,000	1,000	0,500	0,667	1,000	1,000	1,000
Xã hội - Nhân văn	1,000	1,000	1,000	0,600	0,750	0,667	0,600	0,750	0,667
Kinh tế	1,000	1,000	1,000	0,900	0,818	0,857	1,000	0,545	0,706
Tỷ lệ chính xác trung bình			91,2%			80,9%			76,5%

Tuy nhiên, không phải lĩnh vực nào cũng được phân lớp tốt. Xét giải thuật có tỷ lệ phân lớp chính xác tốt nhất (SVM) thì vẫn tồn tại một số lĩnh vực phân loại chưa tốt, chẳng hạn lĩnh vực "Môi trường" và "Công nghệ sinh học" có $F_1 < 67\%$ so với các lĩnh vực khác ($F_1 > 85\%$). Lý do là độ bao phủ của hai lĩnh vực này khá thấp (≤ 0.5), tức là tỷ lệ bài báo được dự đoán đúng hai lĩnh vực này so với thực tế chưa cao. Điều này được lý giải là do có sự chồng lấn (overlap) về lĩnh vực của bài báo, nghĩa là một bài báo vừa thuộc lĩnh vực này nhưng có thể vừa thuộc lĩnh vực khác (ví dụ, "công nghệ thực thẩm" và "công nghệ sinh học" là hai lĩnh vực khá tương đồng). Các lĩnh vực còn lại có độ chính xác và bao phủ khá cao, chứng tỏ các lĩnh vực này tương đối khác biệt với các lĩnh vực còn lại.

5 KẾT LUẬN

Trong bài viết này, giải pháp tự động phân loại bài báo khoa học sử dụng các giải thuật máy học đề xuất nhằm hỗ trợ các tác giả, ban biên tập tiết kiệm thời gian và công sức khi xử lý bài viết trên hệ thống. Các bước tiền xử lý dữ liệu là rất quan trọng để có được tập dữ liệu chuẩn tiến hành chạy các giải thuật này. Kết quả thực nghiệm cho thấy giải thuật phân loại SVM cho kết quả phân loại tốt hơn nhiều so với hai giải thuật Naïve Bayes và kNN.

Kết quả nghiên cứu này cho thấy các kỹ thuật máy học có thể dễ dàng áp dụng vào các bài toán phân loại, lọc và tìm kiếm nội dung. Với mô hình đề xuất, việc rút trích thông tin và tự động phân loại bài báo khoa học khi tác giả gửi đăng trên các tạp chí hoàn toàn khả thi. Mô hình này cũng sử dụng cho việc phân loại, xác định chủ đề bài viết ở các hội thảo khoa học. Thực nghiệm trên các tập dữ liệu lớn hơn sẽ được tiếp tục thực hiện trong tương lai.

TÀI LIỆU THAM KHẢO

Aggarwal, C. C. and Zhai, C., 2012. *In*: Aggarwal, C. C. and Zhai, C. (Eds.). *Mining Text Data*. Springer US. Boston, MA, 163-222.

Bijaksana, M. A., Li, Y. and Algarni, A., 2013. A Pattern Based Two-Stage Text Classifier. *In*: Perner P. (eds). Machine Learning and Data Mining in Pattern Recognition. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 169-182.

Boser, B. E., Guyon, I. M. and Vapnik, V. N., 1992, A training algorithm for optimal margin classifiers. In. Proceedings of the fifth annual workshop on Computational learning theory, Pittsburgh, Pennsylvania, USA. ACM. 130401, 144-152.

Burges, C. J. C., 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*. *2* (2): 121-167.

Chakrabarti, S., 2003. *Mining the Web: Discovering Knowledge from Hypertext Data*.

Chen, J., Huang, H., Tian, S. and Qu, Y., 2009. Feature selection for text classification with Naïve Bayes. *Expert Syst. Appl. 36* (3): 5432-5435.

Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine learning*. *20* (3): 273-297.

Dumais, S., Platt, J., Heckerman, D. and Sahami, M., 1998, Inductive learning algorithms and representations for text categorization. In. Proceedings of the seventh international conference on Information and knowledge management, Bethesda, Maryland, USA. ACM. 288651: 148-155.

George, H. J. and Pat, L., 1995, Estimating continuous distributions in Bayesian classifiers. In. Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, Montréal,

Qué, Canada. Morgan Kaufmann Publishers Inc. 2074196: 338-345.

Haddoud, M., Mokhtari, A., Lecroq, T. and Abdeddaïm, S., 2016. Combining supervised term-weighting metrics for SVM text classification with extended term representation. *Knowledge and Information Systems*. *49* (3): 909-931.

Li, Y., Zhang, L., Xu, Y., Yao, Y., Lau, R. Y. K. and Wu, Y., 2017. Enhancing Binary Classification by Modeling Uncertain Boundary in Three-Way Decisions. *IEEE Transactions on Knowledge and Data Engineering*. *29* (7): 1438-1451.

Liu, B., Dai, Y., Li, X., Lee, W. S. and Yu, P. S., 2003. Building text classifiers using positive and unlabeled examples. Third IEEE International Conference on Data Mining, pp. 179-186.

McCallum, A. and Nigam, K., 1998. A comparison of event models for naive bayes text classification. AAAI-98 workshop on learning for text categorization. Citeseer, pp. 41-48.

Mitchell, T., 1997. Machine Learning, McGraw-Hill Higher Education. *New York*.

Nguyễn Thị Minh Huyền, Vũ Xuân Lương và Lê Hồng Phương., 2010. *VnTokenizer*, accessed on July 15, 2019. Available from https://sourceforge.net/projects/vntokenizer/.

Perone, C. S., 2013. *Machine Learning :: Cosine Similarity for Vector Space Models (Part III)*, accessed on July 20, 2019. Available from http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/.

Sebastiani, F., 2002. Machine learning in automated text categorization. *ACM Comput. Surv. 34* (1): 1-47.

Tan, P.-N., Steinbach, M. and Kumar, V., 2006. Data Mining Introduction. Bei Jing: The people post and Telecommunications Press.

Thaoroijam, K., 2014. A Study on Document Classification using Machine Learning Techniques. *IJCSI International Journal of Computer Science*. *11:* 217-222

Trần Cao Đệ và Phạm Nguyên Khang, 2012. Phân loại văn bản với máy học véc-tơ hỗ trợ và cây quyết định. *Tạp chí Khoa học Trường Đại học Cần Thơ*. *21a*: 52-63.

Trần Thị Thu Thảo và Vũ Thị Chinh, 2012. Xây dựng hệ thống phân loại tài liệu tiếng Việt. Báo cáo nghiên cứu khoa học. Trường Đại học Lạc Hồng. Đồng Nai.

Tsai, C.-H., 2000. *MMSEG: A Word Identification System for Mandarin Chinese Text Based on Two Variants of the Maximum Matching Algorithm*, accessed on July 22, 2019. Available from http://technology.chtsai.org/mmseg/.

Yang, Y. and Liu, X., 1999. A re-examination of text categorization methods. Sigir, pp. 99.

Yang, Y. and Pedersen, J. O., 1997, A Comparative Study on Feature Selection in Text Categorization. In. Proceedings of the Fourteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. 657137: 412-420.

Zhang, L., Li, Y., Sun, C. and Nadee, W., 2013. Rough Set Based Approach to Text Classification. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), pp. 245-252.