ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

- - - - - - - - - - - - - - - - - -

BÁO CÁO THỰC TẬP

NGÀNH KHOA HỌC MÁY TÍNH

ĐỀ TÀI:

Học Máy Trong Trích Xuất Quan Hệ Sử Dụng Giám Sát Yếu

Giảng viên hướng dẫn: TS. Nguyễn Bá Đạt

Sinh viên: Lê Ngọc Tuấn Khang
Mã sinh viên: 15021039
Lớp: K60-CAC

Hà Nội, tháng 10 năm 2018

LỜI CẢM ƠN

Trước nhất, tôi xin chân thành cảm ơn giảng viên hướng dẫn - TS. Nguyễn Bá Đạt về những lời khuyên vô giá và những chỉ bảo tận tâm trong ba tháng thực tập vừa qua. Thầy đã gợi mở nhiều kiến thức mới và đem lại cho tôi một cái nhìn khác về phong cách nghiên cứu trong ngành học của mình. Cùng với đó, tôi muốn gửi lời cảm ơn tới những đàn anh tại nerd.vn, các anh đã giúp đỡ tôi rất nhiều trong đợt thực tập này. Cuối cùng, cảm ơn Khoa Công Nghệ Thông Tin đã tạo điều kiện cho tôi có một trải nghiệm về nghiên cứu khi còn là sinh viên, làm tiền đề cho những bước tiến của tôi sau khi ra trường.

Mục lục

1 Giới thiệu

Trong thời đại internet, dữ liệu văn bản từ các nguồn như báo điện tử, blog, diễn đàn, mạng xã hội liên tục được cập nhật và chứa trong đó một lượng thông tin khổng lồ. Tự động *trích xuất thông tin* (Information Extraction) có cấu trúc từ nguồn dữ liệu không có cấu trúc này là một bài toán quan trọng, là nền tảng của *hệ tri thức* (Knowledge Base) như **YAGO** ([1], [5]), **DBpedia** [2], **Wikidata** [3] và góp phần không nhỏ vào thành công của các *hệ thống hỏi đáp tự động* (Question-answering System), *công cụ tìm kiếm* (Information Retrieval) hay các bài toán xử lí ngôn ngữ tự nhiên khác. [4]

Trích xuất quan hệ (Relation Extraction) là bài toán con của trích xuất thông tin, nhằm xác định tự động *quan hệ* (Relation) giữa các *thực thể định danh* (Named Entity) trong một đoạn văn bản. Thực thể định danh ám chỉ những đối tượng có thực như người (ví dụ: Phạm Bảo Sơn), địa điểm (ví dụ: Hà Nội), hay tổ chức (ví dụ: Trường Đại học Công Nghệ), v.v. Một số quan hệ phổ biến giữa các thực thể xuất hiện trong các hệ tri thức là `nơi sinh`, `học vấn`, `vợ chồng`, v.v. Một cặp thực thể định danh cùng quan hệ giữa chúng được gọi là một tri thức (Fact), như cặp (Phạm Bảo Sơn, Hà Nội) và quan hệ `nơi sinh`.

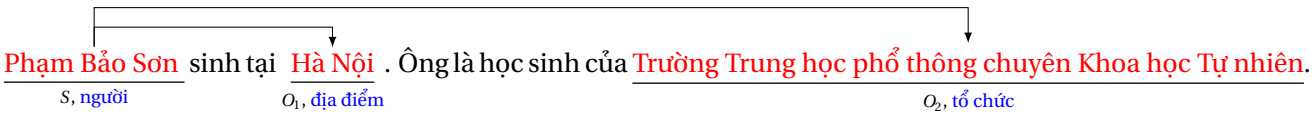
Dựa trên giả thiết một tập các quan hệ đã được biết trước, báo cáo này giải quyết bài toán trích xuất một quan hệ đơn lẻ. Cụ thể, với một đoạn văn bản và một quan hệ đã xác định (ví dụ quan hệ `nơi sinh`), bài toán yêu cầu xác định đoạn văn bản này có chứa tri thức về `nơi sinh` hay không.

Trong báo cáo này, bài toán trên được tiếp cận dưới dạng một bài toán *phân lớp nhị phân* (Binary Classification), ở đó một mô hình phân lớp được huấn luyện, nhận đầu vào là một đoạn văn bản chứa 2 thực thể định danh, và đầu ra là một trong hai nhãn (0 hoặc 1) ám chỉ quan hệ giữa 2 thực thể này là có phải `nơi sinh` hay không.

2 Mô hình hóa bài toán

Một *tri thức* được biểu diễn dưới dạng bộ ba (S, R, O), trong đó S, O là các thực thể định danh, cụ thể S là *chủ thể* (Subject), O là *đối tượng* (Object). Bài toán được đặt ra là: Cho một đoạn văn bản D và quan hệ R, xác định tập tri thức về quan hệ R: $F = \{(S, R, O) | S, O \in E\}$, với E là tập các thực thể trong D.

Ở ví dụ dưới, D = "*Phạm Bảo Sơn sinh tại Hà Nội. Ông là học sinh của Trường Trung học phổ thông chuyên Khoa học Tự nhiên.*", chủ thể S = Phạm Bảo Sơn, các đối tượng O_1 = Hà Nội và O_2 = Trường Trung học phổ thông chuyên Khoa học Tự nhiên. Có hai tri thức trong văn bản này: một tri thức về quan hệ `nơi sinh` là $f_1 = (S, \texttt{nơi sinh}, O_1)$ và một tri thức về `học vấn` $f_2 = (S, \texttt{học vấn}, O_2)$. Nếu $R = \texttt{nơi sinh}$, $F = \{f_1\}$ là tập tri thức cần thu được.



3 Kiến thức nền tảng

3.1 Nhận dạng thực thể định danh

Để thu được quan hệ giữa các thực thể định danh từ một văn bản, trước hết ta phải xác định các thực thể xuất hiện trong đó. *Nhận dạng thực thể định danh* (Named Entity Recognition - NER) là bài toán định vị và phân loại thực thể trong văn bản thành về các nhãn định trước như người, địa điểm, tổ chức, v.v. Ở ví dụ dưới, trong câu "*Phạm Bảo Sơn sinh tại Hà Nội.*", thực thể Phạm Bảo Sơn có nhãn người, và Hà Nội có

nhãn địa điểm.

Phạm Bảo Sơn (người) sinh (O) tại (O) Hà Nội (địa điểm) . (O)

Ông (O) là (O) của (O) Trường (tổ chức) Trung học (tổ chức) phổ thông (tổ chức) chuyên (tổ chức) chuyên (tổ chức) Khoa học (tổ chức) Tự nhiên (tổ chức) . (O)

Nhận diện thực thể định danh là một bài toán quan trọng trong lĩnh vực xử lí ngôn ngữ tự nhiên và là bước đầu cho mọi bài toán trích xuất thông tin. NER thường được mô hình hóa thành bài toán *ghi nhãn chuỗi* (Sequence Labeling), ở đó mỗi từ trong văn bản được gán một nhãn trong tập nhãn định trước, gồm có nhãn thực thể (người, địa điểm, tổ chức v.v.) và nhãn trống O. Ghi nhãn chuỗi, như đa số bài toán học máy khác, có thể tiếp cận theo hai cách: trích xuất đặc trưng thủ công (huấn luyện bằng MEMM/CRF) hoặc trích xuất đặc trưng tự động (huấn luyện bằng mạng nơ ron, như bi-LSTM) [6].

3.2 Đặc trưng

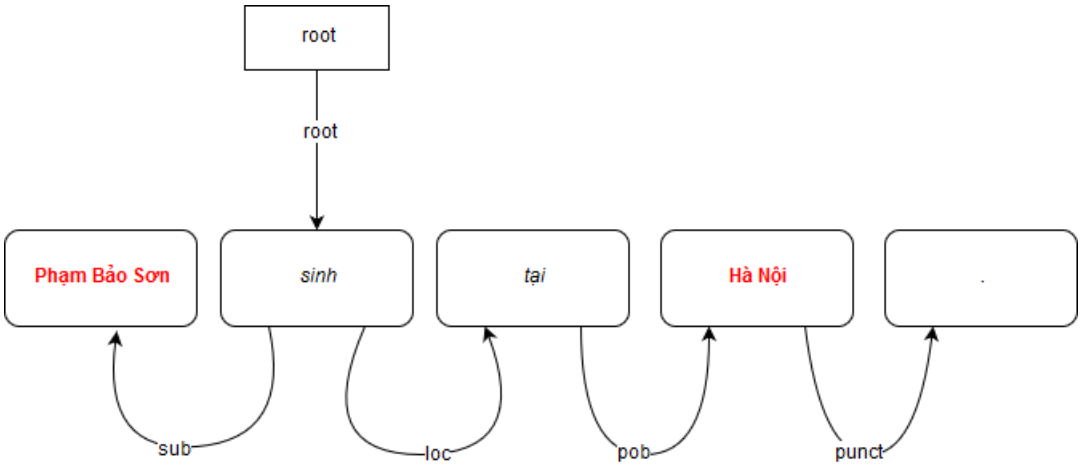
Phần này liệt kê những đặc trưng phổ biến của văn bản. Trích xuất đặc trưng là bước quan trọng trong mọi bài toán học máy, nhằm biểu diễn các đặc tính của một sự vật (ở đây là một đoạn văn bản) dưới dạng tập các giá trị để hỗ trợ việc tính toán và mô hình hóa.

3.2.1 Đặc trưng từ vựng

- *n-gram*: tập *n* từ liên tiếp trong đoạn văn bản.
- *Túi từ* (Bag-of-words): tập hợp tất cả các từ trong văn bản.
- *Từ loại* (POS-tag): như danh từ (N), cụm danh từ (Np), động từ (V), giới từ (E), dấu câu (CH), ...

3.2.2 Đặc trưng cú pháp

- Dãy từ loại của các từ giữa hai thực thể.
- Đường đi trên *Dependency Parsing* của văn bản. Một *Dependency Parser* [7] phân tích cấu trúc ngữ pháp của một văn bản, biểu diễn mối quan hệ phụ thuộc giữa các từ trong đó (hình 1).



Hình 1: Ví dụ về Dependency Parsing cho văn bản "*Phạm Bảo Sơn sinh tại Hà Nội.*"

Bảng 1 cho một số ví dụ cụ thể về các đặc trưng trình bày ở trên.

	Phạm Bảo Sơn sinh tại Hà Nội.	Ông là học sinh của Trường Trung học phổ thông chuyên Khoa học Tự nhiên.
Nhận dạng thực thể định danh	Phạm Bảo Sơn: người, Hà Nội: địa điểm	Trường Trung học phổ thông chuyên Khoa học Tự nhiên: tổ chức
Tách từ	[*Phạm_Bảo_Sơn, sinh, tại, Hà_Nội, .*]	[*Ông, là, học_sinh, của, Trường, Trung_học, phổ_thông, chuyên, Khoa_học, Tự_nhiên, .*]
2-gram quanh thực thể	Phạm Bảo Sơn: [*null, null, sinh, tại*], Hà Nội: [*sinh, tại, ., null*]	Trường Trung học phổ thông chuyên Khoa học Tự nhiên: [*học_sinh, của, ., null*]
Túi từ	[*Phạm_Bảo_Sơn, sinh, tại, Hà_Nội, .*]	[*Ông, là, học_sinh, của, Trường, Trung_học, phổ_thông, chuyên, Khoa_học, Tự_nhiên, .*]
Từ loại	[Np, V, E, CH]	[Nc, V, N, E, N, N, N, V, N, A, CH]
Dãy từ loại giữa hai thực thể	[V]	[V, N, E]
Đường đi trên Dependency Parsing	[(sub, 2), (root, 0), (loc, 2), (pob, 3), (punct, 4)]	[(sub, 2), (root, 0), (pob, 2), (adv, 3), (nmod, 3), (nmod, 4), (nmod, 5), (nmod, 6), (nmod, 7), (nmod, 8), (punct, 9)]

Bảng 1: Ví dụ về các đặc trưng của văn bản

3.3 Mô hình học máy

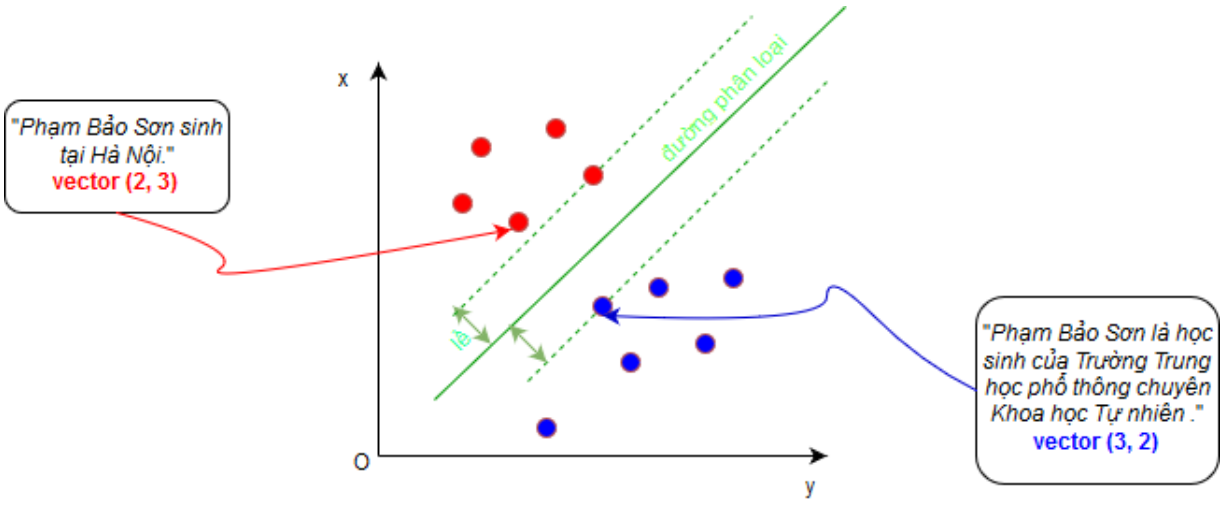
Nhiều thuật toán học máy có thể áp dụng để giải quyết bài toán phân lớp. Những thuật toán này đều có mục tiêu chung là xây dựng một mô hình cho dữ liệu tốt nhất để phân loại từ một tập dữ liệu đã được gán nhãn (gọi là tập *dữ liệu huấn luyện*). Trong báo cáo này, ba thuật toán phổ biến được sử dụng là *Support Vector Machine* [8], *Naive Bayes* và *cây quyết định* [9]. Mỗi thuật toán đều có điểm mạnh và điểm yếu riêng, phù hợp với từng loại dữ liệu và yêu cầu bài toán.

3.3.1 Support Vector Machine

Support Vector Machine (SVM) là một mô hình học máy *có giám sát* thường được dùng trong các bài toán phân loại nhị phân. Ý tưởng cơ bản của SVM là: coi mỗi dữ liệu huấn luyện ứng với một điểm trong không gian N chiều, mô hình sẽ tìm một siêu phẳng $N-1$ chiều tốt nhất chia đôi tập điểm sao cho các dữ liệu cùng lớp thuộc cùng một phần không gian. Tốt nhất ở đây theo nghĩa siêu phẳng có lề lớn nhất, với lề là khoảng cách từ điểm gần nhất (ở mỗi lớp) tới siêu phẳng.

Hình 2 là thuật toán SVM áp dụng cho bài toán phân lớp quan hệ nơi sinh. Mỗi văn bản được biểu diễn bằng một điểm véc tơ 2 chiều, văn bản thuộc lớp "có quan hệ `nơi sinh`" có màu đỏ, văn bản thuộc lớp còn lại có màu xanh. Đường phân lớp của thuật toán SVM là đường xanh lá chia mặt phẳng thành hai phần, mỗi phần tương ứng với một lớp. Lề là khoảng cách từ đường này đến điểm đỏ/xanh gần nhất.

Một văn bản có thể được chuyển về dạng véc tơ bằng cách lựa chọn các đặc trưng của nó và biểu diễn mỗi đặc trưng bởi một/nhiều giá trị số. Ví dụ, đặc trưng từ vựng có thể biểu diễn bởi chỉ số (id) của từ trong từ điển hoặc một véc tơ nhị phân (*one-hot vector*) có độ dài bằng từ điển, có giá trị 1 tại vị trí của từ đang xét trong từ điển và 0 ở tất cả vị trí còn lại.



Hình 2: Ví dụ về SVM cho mặt phẳng

3.3.2 Naive Bayes

Naive Bayes là một mô hình phân lớp xác suất dựa trên định lí Bayes và một giả định "ngây thơ" về tính độc lập xác suất. Cụ thể, nếu một dữ liệu được biểu diễn bởi một véc tơ đặc trưng $X = [x_1, x_2, ..., x_n]$, mỗi dữ liệu thuộc một trong các lớp $C = \{c_1, c_2, ..., c_k\}$, thì Naive Bayes sẽ tính các xác suất $P_i = P(c_i|X)$, với $i = 1, .., k$ từ tập dữ liệu huấn luyện, và gán dữ liệu mới X vào lớp i có xác suất P_i lớn nhất:

$$X \in c_k, \text{ ở đó } k = \arg\max_i \ P(c_i|X) \tag{1}$$

Định lí Bayes cho biết:

$$P(c_i|X) = \frac{P(c_i)P(X|c_i)}{P(X)} \tag{2}$$

Do đó:

$$\arg\max_i \ P(c_i|X) = \arg\max_i \ P(c_i)P(X|c_i) = \arg\max_i \ P(c_i)P(x_1, x_2, ..., x_n|c_i) \tag{3}$$

Giả thiết rằng các đặc trưng $\{x_1, x_2, ..., x_n\}$ đôi một độc lập với nhau, ta có:

$$P(x_1, x_2, ..., x_n|c_i) = P(x_1|c_i)P(x_2|c_i)...P(x_n|c_i) \tag{4}$$

Từ dữ liệu huấn luyện, ta có thể tính được các giá trị $P(c_i)$ và $P(x_j|c_i)$ $(j = 1..n, i = 1..k)$, từ đó tính được lớp cho xác suất cao nhất khi biết dữ liệu X.

3.3.3 Cây quyết định

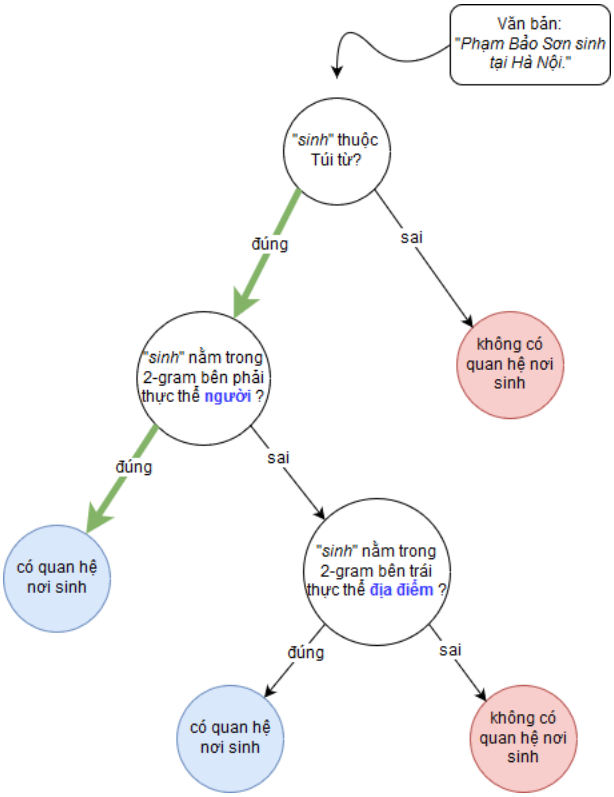
Cây quyết định (Decision Tree) là một đồ thị dạng cây của các quyết định và hậu quả của nó. Mỗi nút (trừ lá) trên cây biểu diễn một đánh giá, một cạnh nối từ nút cha đến nút con biểu diễn giá trị có thể của nút cha, và lá tương ứng với kết quả quyết định cuối cùng.

Hình 3 cho một ví dụ về cây quyết định của bài toán kiểm tra quan hệ `noi sinh`, ở đó đầu vào là một đoạn văn bản, đầu ra là quyết định văn bản đó có chứa quan hệ `noi sinh` hay không. Những nút trắng là các đánh giá nhị phân, nhận kết quả đúng hoặc sai; các nút màu là lá, ứng với kết quả cuối cùng (nhãn "có" hoặc "không có") .

Để mô hình hóa cây quyết định, mỗi đánh giá được coi là một đặc trưng $f_i \in F$ (F là tập đặc trưng của

đầu vào), nhận các giá trị số $v = \{v_1, v_2, ..., v_k\}$. Xây dựng cây quyết định là bài toán xây dựng một cây có các nút là f_i $(f_i \in F)$, các cạnh nối từ f_i tương ứng với các giá trị (hoặc tập các giá trị) trong v, và lá ứng với kết quả quyết định. Ở cây quyết định trong hình 3, ta có đặc trưng f_1 = ("*sinh*" ∈ Túi từ ?) nhận giá trị $v = \{0, 1\}$ ứng với {sai, đúng}.

Huấn luyện cây quyết định là tìm ra cây quyết định phù hợp nhất với một tập dữ liệu đã được gán nhãn. Điều này dẫn đến việc lựa chọn hợp lí thứ tự đặc trưng làm nút trên cây và các giá trị của đặc trưng đó trên các cạnh. Một số độ đo phổ biến cho quá trình chọn lựa có thể kể đến *độ lợi thông tin* (Information Gain) trong thuật toán *ID3, tỉ lệ độ lợi thông tin* (Gain Ratio) trong *C4.5* và *chỉ số Gini* (Gini Index) trong thuật toán *CART*.



Hình 3: Ví dụ về cây quyết định

4 Thực nghiệm

Chương này nêu các bước được tiến hành để giải quyết bài toán đề ra. Có 5 giai đoạn chính, đó là:

- Giai đoạn 1: Thu thập dữ liệu
- Giai đoạn 2: Gán nhãn dữ liệu
- Giai đoạn 3: Trích xuất đặc trưng
- Giai đoạn 4: Huấn luyện mô hình học máy
- Giai đoạn 5: Thử mô hình trên dữ liệu mới

Một số chi tiết sẽ được đề cập dưới đây.

Dữ liệu Tập dữ liệu văn bản được lấy từ các trang báo dantri.com.vn, baomoi.com bằng kĩ thuật khớp *biểu thức chính quy* (Regular Expression); tổng cộng gồm 3842 văn bản.

Mỗi câu được chạy qua bộ nhận dạng thực thể định danh để xác định tập thực thể, và mỗi cặp thực thể trong tập này cùng văn bản cho ta một dữ liệu.

Những dữ liệu được dùng phương pháp giám sát yếu đã đề cập để gán nhãn, thu được 1589 nhãn đúng (nghĩa là có quan hệ `nơi sinh` giữa cặp thực thể trong câu) và 5249 nhãn sai. Để khắc phục sự mất cân bằng về nhãn, tập dữ liệu có nhãn sai được chọn ngẫu nhiên cho đến khi đủ 1589 dữ liệu (phương pháp *down-sampling*) - chính bằng số dữ liệu có nhãn đúng. Như vậy, bộ dữ liệu huấn luyện gồm 3178 dữ liệu, cân bằng số lượng thuộc 2 nhãn.

Bộ *dữ liệu kiểm thử* gồm 687 câu lấy từ các trang báo và Wikipedia, được gán nhãn bằng tay và kiểm tra chéo.

Giám sát yếu Dữ liệu sẽ được gán nhãn bằng phương pháp *giám sát yếu*, cụ thể bao gồm:

- **Kiến thức về miền**: Những văn bản có chứa các cụm như "sinh tại", "quê ở", "đến từ", ... sẽ là một văn bản nói về `nơi sinh`.
- **Giám sát từ xa**:
  - Nếu trong cơ sở tri thức cặp (Phạm Bảo Sơn, Hà Nội) có quan hệ `nơi sinh`, thì những văn bản cùng chứa 2 thực thể Phạm Bảo Sơn và Hà Nội sẽ chứa quan hệ `nơi sinh`.
  - Nếu một văn bản chứa cặp thực thể không có mối quan hệ `nơi sinh` trong cơ sở tri thức, văn bản đó sẽ KHÔNG nói về `nơi sinh`.

Trích xuất đặc trưng Các đặc trưng được dùng cụ thể như sau:

- Khoảng cách (số từ) D giữa hai thực thể.
- Giá trị S thể hiện hai thực thể có cùng nằm trong một câu hay không (chú ý rằng, một văn bản có thể gồm nhiều câu).
- Phiên bản mở rộng của n-gram: cụ thể, xung quanh một thực thể, ta sẽ bỏ qua stopword, dấu câu và những từ có cùng nhãn NER với nó, rồi sau đó mới xét đến n-gram. Cách làm này dựa trên giả định rằng các từ có cùng nhãn NER, chỉ cách nhau bởi các dấu câu hay stopword, sẽ có vai trò như nhau trong một quan hệ nào đó. Do đó, cách lấy n-gram như trên sẽ giúp giảm nhiễu trong các đặc trưng.
  Trong thực nghiệm, $n = 2$ được chọn, và ta xét các 2-gram ở hai phía của 2 thực thể. Do đó véc tơ đặc trưng sẽ có dạng $[L_{1s}, L_{2s}, R_{1s}, R_{2s}, L_{1o}, L_{2o}, R_{1o}, R_{2o}]$

Vậy véc tơ đặc trưng cuối cùng của một văn bản là: $[D, S, L_{1s}, L_{2s}, R_{1s}, R_{2s}, L_{1o}, L_{2o}, R_{1o}, R_{2o}, D]$.
Ví dụ, văn bản *"Phạm Bảo Sơn sinh tại Hà Nội."* có véc tơ đặc trưng là **[2, 1, 0, 0, 43256, 2975, 43256, 2975, 71327, 0]**; trong đó **2** là khoảng cách giữa Phạm Bảo Sơn và Hà Nội, **1** ám chỉ hai thực thể này cùng nằm trong một câu, **0** là id của từ "*null*" trong từ điển, **43256** là id của "*sinh*", **2975** là id của "*tại*" và **71327** là id của ".".

Cài đặt Các thuật toán SVM, Naive Bayes và cây quyết định được cài đặt sử dụng thư viện scikit-learn [10] của python. Trích xuất đặc trưng được thực hiện trong ngôn ngữ Java, sử dụng bộ NLP-toolkit của NERD.vn [11]. Source code của báo cáo có thể tìm thấy tại https://github.com/lntk/Birthplace-Relation-Extraction.
Về thuật toán và tham số, thuật toán Soft Margin SVM được sử dụng, sử dụng hàm nhân RBF (Radial basis function) và $C = 1.0$. Naive Bayes mô hình dữ liệu bằng phân phối Gaussian. Cây quyết định dùng chỉ số Gini để lựa chọn đặc trưng.

Độ tin cậy (Confidence Score) Nếu trích xuất quan hệ được sử dụng để cập nhật hệ tri thức, *độ tin cậy* (độ chính xác, độ đúng đắn) của việc phân loại một quan hệ là cần thiết, do thêm quan hệ sai vào hệ tri thức là một điều cấm kị. Với mô hình SVM, độ tin cậy khi phân lớp một điểm dữ liệu có thể được tính bởi khoảng của nó đến siêu phẳng phân lớp. Với Naive Bayes, xác suất $P_i = P(c_i|X)$ (đã được chuẩn hóa) có thể được dùng để biểu thị độ đúng đắn. Trong cây quyết định, ta không có cách nào tính được giá trị này.

5 Kết quả

Các độ đo thường dùng để đánh giá một mô hình phân lớp là *độ chính xác* (Precision), *độ hồi tưởng* (Recall) và *F1-score*. Với các giá trị *true positive* (TP), *false positive* (FP), *true negative* (TN), *false negative* (FN) được định nghĩa trong bảng 2. Các độ đo trên được tính như sau:

Tỉ lệ đúng trên những dữ liệu được gán nhãn **1**:

$$\text{precision} = \frac{TP}{TP + FP} \tag{5}$$

Tỉ lệ những dữ liệu nhãn **1** được dự đoán đúng:

$$\text{recall} = \frac{TP}{TP + FN} \tag{6}$$

Đánh giá đồng thời **precision** và **recall** sử dụng:

$$\text{f1-score} = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} \tag{7}$$

	nhãn đúng = 1	nhãn đúng = 0
dự đoán = 1	TP	FP
dự đoán = 0	FN	TN

Bảng 2: Định nghĩa TP, FP, TN, FN

Bảng 3 là kết quả khi áp dụng các thuật toán học máy cho bài toán phân lớp quan hệ nơi sinh. SVM cho độ chính xác cao (1.0), nhưng độ hồi tưởng thấp (0.08), do đó phù hợp khi tính đúng đắn của quan hệ thu về được đặt lên hàng đầu (ví dụ khi trích xuất quan hệ để thêm vào hệ tri thức). Ngược lại, Naive Bayes đem đến độ hồi tưởng cao, phù hợp với mục đích thu lại càng nhiều quan hệ càng tốt (tính chính xác của quan hệ có thể được kiểm tra ở những bước sau). Cuối cùng, cây quyết định cho hai giá trị chính xác và hồi tưởng khá cân bằng ở mức trung bình.

	Precision	Recall	F1-score
SVM	**1.0**	0.08	0.10
Cây quyết định	0.52	0.47	0.49
Naive Bayes	0.37	**0.84**	**0.51**

Bảng 3: Kết quả của các thuật toán

6 Hướng phát triển

Báo cáo này đưa ra cách tiếp cận đơn giản cho bài toán trích xuất quan hệ sử dụng một số thuật toán học máy cơ bản cùng kĩ thuật giám sát yếu để gán nhãn dữ liệu. Trích xuất quan hệ vẫn là một vấn đề mở, và có nhiều cách cải tiến những phương pháp đề cập ở trên như:

- **Thu thập nhiều dữ liệu hơn (mở rộng crawler)**: Phương pháp giám sát yếu sử dụng những giả thiết dựa trên kinh nghiệm, nên rất dễ sinh ra những dữ liệu *nhiễu* (bị gán nhãn sai). Điều này có thể được khắc phụ nếu một lượng lớn dữ liệu được cung cấp, khi đó lượng nhiễu trở nên không đáng kể, và mô hình học máy sẽ học được những đặc trưng chung từ lượng lớn dữ liệu đó.

- **Cải tiến kĩ thuật giám sát yếu**: Đây cũng là một cách tiếp cận với mục đích giảm nhiễu gây ra bởi giám sát yếu. Cách đơn giản nhất là tăng cường những kĩ thuật gán nhãn (như sử dụng *bộ phân lớp yếu* (boosting), hay *crowdsourcing*, ...). Hơn thế, một bộ học có thể được huấn luyện để chọn ra những kĩ thuật có khả năng gán nhãn đúng cho một dạng câu cụ thể, từ đó khắc phục vấn đề gán nhãn sai [12].

- **Sử dụng *chọn lọc đặc trưng* (Feature selection)**: Không phải đặc trưng nào cũng có lợi cho việc phân lớp. Việc loại bỏ đặc trưng bằng tay sẽ tiềm ẩn khả năng bỏ qua những đặc trưng quan trọng, trong khi sử dụng quá nhiều đặc trưng có thể dẫn đến khó khăn (và giảm chất lượng) của bộ học. Phương pháp chọn lọc đặc trưng lấy ra từ tập đặc trưng một tập con giúp cho việc phân lớp dữ liệu dễ dàng nhất, nhờ đó cải thiện kết quả.

- **Cải tiến mô hình học máy**: Những mô hình học máy cổ điển như SVM, cây quyết định hay Bayes ngây thơ cần những đặc trưng được định nghĩa trước, và do đó phụ thuộc vào quyết định của con người. Một số mô hình học máy có khả năng tự động trích xuất đặc trưng như *mạng nơ ron nhân tạo* (Artificial Neural Network), và cải tiến của nó là những mạng như *mạng hồi quy* [13] (Recurrent Neural Network) hay *Long short-term memory* [14] có thể trích xuất đặc trưng từ mối liên hệ giữa những phần tử của một dãy (như các từ trong một văn bản), sẽ làm cải thiện đáng kể kết quả của bộ phân lớp, với điều kiện một lượng dữ liệu cực lớn được cung cấp.

Tham khảo

[1] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

[2] https://wiki.dbpedia.org/

[3] https://www.wikidata.org/

[4] Bach, Nguyen, and Sameer Badaskar. "A review of relation extraction." Literature review for Language and Statistics II 2 (2007).

[5] Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum. "Yago: a core of semantic knowledge." Proceedings of the 16th international conference on World Wide Web. ACM, 2007.

[6] Jurafsky, Dan, and James H. Martin. Speech and language processing. Vol. 3. London: Pearson, 2014.

[7] https://nlp.stanford.edu/software/nndep.html

[8] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297

[9] Breiman, Leo. "Bagging predictors." Machine learning 24.2 (1996): 123-140.

[10] http://scikit-learn.org/

[11] https://www.nerd.vn/

[12] https://hazyresearch.github.io/snorkel/

[13] Mikolov, Tomáš, et al. "Recurrent neural network based language model." Eleventh Annual Conference of the International Speech Communication Association. 2010.

[14] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

Ý kiến đánh giá

.....

.....

.....

.....

.....

.....

.....

.....

Điểm số: ..... Điểm chữ: .....

Hà Nội, ngày ... tháng ... năm 2018

Giảng viên đánh giá