




POS Tagging system use Support Vector Machine (SVM)

Created by	 Tiến Đạt Lê Văn
Created time	@March 19, 2024 10:27 PM
Tags	Product

I. Giới thiệu về bài toán POS Tagging

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), việc hiểu ngữ cảnh của một câu là một phần quan trọng để nắm bắt ý nghĩa và cấu trúc của văn bản. Một trong những nhiệm vụ quan trọng nhất của NLP là Part-of-Speech (POS) Tagging, có mục tiêu là gán nhãn loại từ (part-of-speech) cho mỗi từ trong một câu. Bằng cách này, POS Tagging giúp phân biệt các loại từ như danh từ, động từ, tính từ, phó từ, v.v., và cung cấp thông tin về vai trò và mối quan hệ giữa các từ trong câu.

Ví dụ như trong câu "Đạt huấn luyện Robot" thì pos tagging sẽ nhận diện "Đạt" là danh từ (noun), "huấn luyện" là động từ (Verb) và "Robot" là danh từ (noun)

POS Tagging đóng vai trò quan trọng trong nhiều ứng dụng của NLP như phân tích cú pháp, dịch máy, tóm tắt văn bản, trích xuất thông tin, v.v. Bằng cách hiểu được loại từ của mỗi từ trong câu, chúng ta có thể giải quyết hiệu quả các nhiệm vụ phức tạp như xây dựng cấu trúc cú pháp của câu, phân tích ngữ cảnh, và nắm bắt ý nghĩa của văn bản.

II. Bảng từ loại

STT	Nhãn	Tên	Ví dụ
1	N	Danh từ	tiếng, nước, thủ đô, nhân dân, đồ đạc, cây cối, chim muông
2	Np	Danh từ riêng	Nguyễn Du, Việt Nam, Hải Phòng, Trường Đại học Bách khoa Hà Nội, Mộc tinh, Hoả tinh, Phật, Đạo Phật
3	Nc	Danh từ chỉ loại	con, cái, đứa, bức
4	Nu	Danh từ đơn vị ¹	mét, cân, giờ, nắm, nhúm, hào, xu, đồng
5	V	Động từ	ngủ, ngồi, cười; đọc, viết, đá, đặt; thích, yêu, ghét, giống, muốn
6	A	Tính từ	tốt, xấu, đẹp; cao, thấp, rộng
7	P	Đại từ	tôi, chúng tôi, hắn, nó, y, đại nhân, đại ca, huynh, đệ
8	L	Định từ ²	mỗi, từng, mọi, cái; các, những, mấy
9	M	Số từ	một, mười, mười ba; dăm, vài, mười; nửa, rưỡi
10	R	Phó từ	đã, sẽ, đang, vừa, mới, từng, xong, rồi; rất, hơi, khi, quá
11	E	Giới từ ³ (kết từ chính phụ)	trên, dưới, trong, ngoài; của, trừ, ngoài, khỏi, ở
12	C	Liên từ (kết từ đẳng lập)	và, với, cùng, vì vậy, tuy nhiên, ngược lại
13	I	Thán từ	ôi, chao, a ha
14	T	Trợ từ, tình thái từ (tiểu từ) ⁴	à, a, á, à, ấy, chắc, chẳng, cho, chứ
15	B	Từ tiếng nước ngoài (hay từ vay mượn)	Internet, email, video, chat
16	Y	Từ viết tắt	OPEC, WTO, HIV
17	S	Yếu tố cấu tạo từ	bất, vô, gia, đa
18	X	Các từ không phân loại được	

III. Support Vector Machine (SVM)

Support Vector Machine (SVM) là một thuật toán học máy sử dụng các mô hình học có giám sát để giải quyết các vấn đề phân loại, hồi quy và phát hiện ngoại lệ phức tạp bằng cách thực hiện các phép biến đổi dữ liệu tối ưu nhằm xác định ranh giới giữa các điểm dữ liệu dựa trên các lớp, nhãn hoặc đầu ra được xác định trước. SVM được áp dụng rộng rãi trong các lĩnh vực như chăm sóc sức khỏe, xử lý ngôn ngữ tự nhiên, ứng dụng xử lý tín hiệu và lĩnh vực nhận dạng giọng nói & hình ảnh.

Các loại thuật toán của SVM

- **Linear SVM:** Khi dữ liệu chỉ có thể phân tách tuyến tính hoàn hảo thì chúng ta có thể sử dụng SVM tuyến tính. Có thể phân tách tuyến tính hoàn hảo có nghĩa là các điểm dữ liệu có thể được phân loại thành 2 lớp bằng cách sử dụng một đường thẳng duy nhất
- **Non-Linear SVM:** Khi dữ liệu không thể phân tách tuyến tính thì chúng ta có thể sử dụng Non-Linear SVM, nghĩa là khi các điểm dữ liệu không thể tách thành 2 lớp bằng cách sử dụng đường thẳng (nếu 2D) thì chúng ta sử dụng một số kỹ thuật nâng cao như kernel thủ thuật để phân loại chúng. Trong hầu hết các ứng dụng trong thế giới thực, chúng tôi không tìm thấy các điểm dữ liệu có thể phân tách tuyến tính, do đó chúng tôi sử dụng thủ thuật kernel để giải quyết chúng.

Cách hoạt động của SVM

1. Tìm mặt phẳng tối ưu

Giả sử chúng ta có một tập dữ liệu đào tạo được biểu diễn trong không gian hai chiều với hai lớp, lớp màu xanh và lớp màu đỏ. Mục tiêu của SVM là tìm ra một siêu phẳng tối ưu (đường thẳng trong trường hợp này) để phân tách hai lớp này một cách tốt nhất.

Siêu phẳng tối ưu sẽ chia không gian thành hai phần, mỗi phần chứa một lớp dữ liệu và cách xa nhau nhất. Đường này được chọn sao cho margin (khoảng cách giữa siêu phẳng và điểm dữ liệu gần nhất từ mỗi lớp, được biểu thị bằng đường màu xanh và đỏ trong hình) là lớn nhất.

2. Margin và Support Vectors:

Margin là khoảng cách từ siêu phẳng tới các điểm dữ liệu gần nhất từ mỗi lớp. Các điểm dữ liệu này được gọi là các vector hỗ trợ. Margin lớn nhất sẽ tạo ra siêu phẳng tối ưu.

Các vector hỗ trợ là các điểm dữ liệu nằm gần nhất với siêu phẳng tối ưu và chúng là những điểm quan trọng nhất trong việc định hình siêu phẳng. Chúng thực sự làm thay đổi vị trí của siêu phẳng khi chúng được thay đổi hoặc xóa bỏ.

3. Hàm Mất Mát (Loss Function) và Tối Ưu Hóa:

Để tìm ra siêu phẳng tối ưu, ta cần tối thiểu hóa hàm mất mát, cũng được gọi là hàm mất mát của SVM:

$$\text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$$

Trong đó:

- w là vector trọng số của siêu phẳng.
- ξ_i là biến lỏng lẻo (slack variable) để xác định độ lỗi cho mỗi điểm dữ liệu.
- C là hằng số regularization, quyết định sự đánh đổi giữa việc tối ưu hóa margin và việc phạt lỗi phân loại.

4. Kernel Trick:

Trong trường hợp dữ liệu không thể phân tách tuyến tính trong không gian hiện tại, chúng ta có thể sử dụng kỹ thuật kernel để ánh xạ dữ liệu vào một không gian đặc trưng cao hơn, nơi mà nó có thể được phân tách tốt hơn. Một số hàm kernel phổ biến bao gồm:

- Linear Kernel: $K(x_i, x_j) = x_i^T x_j$
- Polynomial Kernel: $K(x_i, x_j) = (x_i^T x_j + r)^d$
- Radial Basis Function (RBF) Kernel: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$
- Sigmoid Kernel: $K(x_i, x_j) = \tanh(\alpha x_i^T x_j + c)$

Ưu điểm của SVM:

- Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn.
- Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.
- Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.

Nhược điểm của SVM:

- SVM có thể trở nên rất chậm khi đối mặt với các tập dữ liệu lớn, đặc biệt là khi số chiều của dữ liệu tăng. Việc tìm kiếm siêu phẳng tối ưu đòi hỏi sử dụng các phương pháp tối ưu hóa phức tạp, như là tối ưu hóa bậc hai.
- Hiệu suất của SVM phụ thuộc nhiều vào việc lựa chọn kernel phù hợp và điều chỉnh các tham số tương ứng. Việc lựa chọn sai kernel có thể dẫn


đến hiệu suất kém và overfitting.

- SVM có thể nhạy cảm với các điểm ngoại lai hoặc nhiễu trong dữ liệu, đặc biệt là khi sử dụng các kernel phi tuyến tính. Các điểm ngoại lai có thể trở thành các vector hỗ trợ và ảnh hưởng đến vị trí của siêu phẳng.
- SVM có nhiều tham số cần điều chỉnh như tham số regularization C và các tham số liên quan đến kernel. Điều này có thể đòi hỏi kiến thức chuyên môn sâu về thuật toán và kinh nghiệm trong việc sử dụng SVM hiệu quả.
- SVM có thể không hiệu quả khi đối mặt với các tập dữ liệu không cân bằng, nơi một lớp có số lượng điểm dữ liệu lớn hơn nhiều so với lớp khác. Điều này có thể dẫn đến việc mô hình hướng về lớp có số lượng điểm dữ liệu lớn hơn.

IV. Dataset

- Sử dụng bộ dữ liệu vi_train.txt gồm hơn 15000 câu tiếng việt được thu thập từ internet và có thêm khoảng 150 liên quan đến việc điều khiển nhà thông minh mà không xuất hiện chủ ngữ được sử dụng để huấn luyện Model.
- Bộ dữ liệu vi_test.txt gồm hơn 2000 câu tiếng việt được thu thập từ Internet và có bổ sung hơn 50 câu liên quan đến nhà thông minh được dùng để kiểm thử hệ thống
- Link Dataset:

Postagging Dataset - Google Drive

 <https://drive.google.com/drive/folders/1ayD5zfaD1y2V-CT6hJY5AH1YXSfrKhea?usp=sharing>

V. Data processing

Với 1 câu dữ liệu đầu vào sẽ được phân tách thành từng từ, mỗi từ trong câu sẽ có định dạng như ví dụ dưới đây: "búa/N" sẽ được phân tách thành "búa" và "N" vào các dictionary tiền xử lý tương ứng, với các từ ngữ đặc biệt hơn có 2 từ trở lên dính liền nhau như "thông_dịch_viên" sẽ được phân loại riêng vào các nhóm riêng để xử lý dữ liệu, trong này sẽ có 3 nhóm tương ứng với số gạch nối là 0, 2, 4

Thực hiện tạo ra vector đặc trưng cho từng từ dựa trên nguyên tắc của thuật toán Viterbi trả về vector đặc trưng của từ bao gồm thông tin về phần trăm vị trí của từ trong câu, có viết hoa hay không, và trọng số của các loại POS:

- Đầu tiên, một vector `feat` được khởi tạo với giá trị ban đầu là 1, thường là để biểu diễn trọng số (bias) của mô hình.
- Tính phần trăm của vị trí của từ trong câu (`sentPercent`) và thêm vào vector đặc trưng.
- Nếu từ đầu tiên của câu không viết hoa, hoặc từ không phải là từ đầu tiên trong câu và không viết hoa, thì không có đặc trưng được thêm vào vector.
- Tiếp theo, hàm kiểm tra xem từ hiện tại có trong `wordBank` không. `wordBank` chứa các từ và tập hợp các loại POS (part-of-speech) của từ đó.
- Nếu từ không có trong `wordBank` , hàm sẽ trả về một vector đặc trưng với giá trị tương ứng với từ đầu tiên trong `tagsetDict` và một số loại POS không xuất hiện.
- Nếu từ có trong `wordBank` , hàm tính toán trọng số của các loại POS dựa trên tần suất xuất hiện của chúng trong `wordBank` , và thêm vào vector đặc trưng.

Chuẩn bị dữ liệu để huấn luyện và kiểm thử Pos Tagging:

- Mỗi câu được tách thành các từ. Mỗi từ được tách thành hai phần: từ và nhãn POS được gán vào part, Nếu từ chỉ chứa từ mà không có nhãn POS (tức là `len(parts) == 1`) hoặc nhãn POS không nằm trong `tagsetDict` , thì dừng và tiếp tục với từ tiếp theo.
- Nếu từ có nhãn POS và nằm trong `tagsetDict` , thì nhãn POS đầu tiên của từ (trong trường hợp từ có nhiều nhãn POS) được thêm vào danh sách `y_train (y_test)`.
- Vector đặc trưng cho từ được tạo ra bằng cách gọi hàm `Viterbi_rule_based()` với các đối số là từ hiện tại, vị trí của từ trong câu, độ dài của câu và toàn bộ câu. Kết quả vector đặc trưng được thêm vào danh sách `x_train (x_test)`.
- Tách bớt 1/4 dữ liệu `X_train` và `y_train` vào cho `X_val` và `Y_val` (Validation) để thực hiện huấn luyện model

VI. Tranning model

1. Mục tiêu:

Tạo ra một bộ phân loại có khả năng dự đoán loại từ của mỗi từ trong một câu dựa trên các đặc trưng của từ và ngữ cảnh xung quanh. Cụ thể, mục tiêu là tìm ra siêu phẳng (hyperplane) phân chia tốt nhất giữa các loại từ khác nhau trong không gian đặc trưng của chúng, sao cho khoảng cách giữa siêu phẳng và các điểm dữ liệu của các loại từ là lớn nhất có thể. Mô hình SVM được huấn luyện với mục tiêu tối đa hóa hiệu suất phân loại trên tập dữ liệu huấn luyện.

2. Kiến trúc mô hình:

Sử dụng biến thể của SVM là LinearSVC: Được sử dụng cho các bài toán phân loại nhị phân hoặc phân loại đa lớp với phương pháp 1vs1 (Ở đây thì x là 1 từ, y là từ loại)

3. Đánh giá mô hình:

Mô hình LinearSVC này đã được huấn luyện 05 bước trước khi đánh giá

Accuracy của việc huấn luyện tốt nhất đạt ở 92.60%, của Validation là 92.67%

Accuracy của việc kiểm tra là 87.03% với bảng đánh giá phân lớp ở dưới đây:

	precision	recall	f1-score	support
A	0.77	0.90	0.83	1292
B	1.00	0.14	0.24	22
C	0.83	0.93	0.88	832
E	0.92	0.83	0.87	1643
I	0.33	0.75	0.46	4
L	0.97	0.93	0.95	436
M	0.90	0.99	0.94	747
N	0.95	0.82	0.88	6784
Nc	0.48	0.80	0.60	354
Np	0.75	0.93	0.83	710
Nu	0.82	0.87	0.85	107
P	0.98	0.94	0.96	996
R	0.88	0.82	0.85	1768
S	0.42	0.62	0.50	13
T	0.27	0.53	0.36	73
V	0.85	0.92	0.88	4367
Y	0.00	0.00	0.00	1
accuracy			0.87	20149
macro avg	0.71	0.75	0.70	20149
weighted avg	0.88	0.87	0.87	20149

- Accuracy của việc kiểm tra với Data thật sử dụng trong mô hình nhà thông minh đạt 82.86%

VII. Thực nghiệm với Dataset Treebank

- Bộ dữ liệu này gồm hơn 50.000 từ vựng Tiếng Việt được tách từ các câu thường ngày có thành phần về từ vựng và từ loại của một từ cũng như vị trí trong câu
- Được đánh giá bằng mô hình LinearSVC được huấn luyện 10 bước trước khi đánh giá
- Accuracy của việc huấn luyện tốt nhất đạt ở 93.74%, của Validation là 94.01%
- Accuracy của việc kiểm tra là 77.78% với bảng đánh giá phân lớp ở dưới đây:
-

	precision	recall	f1-score	support
ADJ	0.58	0.82	0.68	504
ADP	0.86	0.76	0.81	729
ADV	0.78	0.90	0.83	886
AUX	0.99	0.69	0.82	344
CCONJ	1.00	0.88	0.94	163
DET	0.90	0.12	0.21	1193
INTJ	0.35	1.00	0.52	6
NOUN	0.85	0.91	0.88	2842
NUM	0.86	0.94	0.90	246
PART	0.52	0.58	0.55	79
PRON	0.94	0.90	0.92	555
PROPN	0.40	0.84	0.54	268
SCONJ	0.76	0.94	0.85	309
SYM	1.00	0.25	0.40	8
VERB	0.73	0.86	0.79	1805
accuracy			0.78	9937
macro avg	0.77	0.76	0.71	9937
weighted avg	0.81	0.78	0.75	9937

- Accuracy của việc kiểm tra với Data thật sử dụng trong mô hình nhà thông minh đạt 64.66%

Bảng so sánh giữa 2 tập dataset

	Training Accuracy	Validation Accuracy	Testing Accuracy	IOT-Testing Accuracy
Vi Dataset	92.60%	92.67%	87.03%	82.86%
Treebank	93.74%	94.01%	77.78%	64.66%