

**ĐẠI HỌC QUỐC GIA TP. HCM**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO TỔNG KẾT**  
**GÁN NHÂN TỪ LOẠI TIẾNG VIỆT**

**Khoa:** Khoa học máy tính

**Môn:** Xử lý ngôn ngữ tự nhiên – CS221.N11.KHCL

**Giáo viên hướng dẫn:** ThS. Nguyễn Trọng Chính

**Tham gia thực hiện**

<b>Họ và tên</b>	<b>MSSV</b>
Phạm Trung Hiếu	19521512
Võ Khoa Nam	19521877
Trịnh Minh Hoàng	19521547

## BẢNG PHÂN CÔNG

Công việc	Hiếu	Nam	Hoàng
Thu thập dữ liệu	33%	33%	33%
Tách từ	30%	40%	30%
Ngữ liệu gán nhãn	25%	40%	35%
Huấn luyện mô hình	55%	25%	20%
Viết báo cáo	33%	33%	33%
Thuyết trình	40%	30%	30%

## **MỤC LỤC**

<b>CHƯƠNG I: TỔNG QUAN .....</b>	<b>4</b>
1. Giới thiệu chung.....	4
2. Giới thiệu bài toán .....	4
<b>CHƯƠNG II: THU THẬP DỮ LIỆU .....</b>	<b>5</b>
1. Nguồn thu thập dữ liệu .....	5
2. Bộ dữ liệu.....	5
<b>CHƯƠNG III: TÁCH TỪ .....</b>	<b>6</b>
1. Lý do tách từ .....	6
2. Tách từ bằng thuật toán Maximum Matching .....	7
3. Tách từ sử dụng thư viện VnCoreNLP .....	8
3.1 Giới thiệu thư viện VnCore NLP .....	8
3.2 Cài đặt.....	8
<b>CHƯƠNG IV: TẠO BỘ NGỮ LIỆU .....</b>	<b>11</b>
1. Tạo bộ ngữ liệu.....	11
1.1. Tách từ thủ công .....	11
1.2. Quy trình tạo ngữ liệu và gán nhãn .....	13
<b>CHƯƠNG V: XÂY DỰNG MÔ HÌNH HIDDEN MARKOV .....</b>	<b>17</b>
1. Hidden Markov .....	17
1.1. Giới thiệu Hidden Markov .....	17
1.2. Ma trận chuyển trạng thái A .....	17
1.3. Ma trận thể hiện B (Emission Matrix).....	19
1.4. Gán nhãn:.....	21
<b>CHƯƠNG VI: ĐÁNH GIÁ .....</b>	<b>24</b>
1. Cách tính accuracy .....	24
2. Kết luận.....	24
<b>CHƯƠNG VII: TÀI LIỆU THAM KHẢO .....</b>	<b>25</b>

# CHƯƠNG I: TỔNG QUAN

## 1. Giới thiệu chung

Part of speech (POS) tagging là một trong những phương pháp quan trọng của xử lý ngôn ngữ tự nhiên, cũng như trong việc hiểu nội dung câu hoặc văn bản. POS là thuật ngữ truyền thống để chỉ các loại từ được phân biệt về mặt ngữ pháp trong một ngôn ngữ. Trong quá trình phát triển chúng ta quen với việc xác định từ loại trong văn bản. Đọc một câu chúng ta có thể xác định rõ từ loại như là danh từ, động từ hoặc tính từ...

Để xác định từ rõ từ loại trong câu thường phức tạp hơn nhiều trong việc ánh xạ các từ qua từ điển. Đó là bởi vì một từ có thể được gán rất nhiều từ loại dựa vào ngữ cảnh của văn bản. Đây gọi là sự nhập nhằng. Thật khó để ta xác định một từ đó thuộc từ loại nào dựa vào một ngữ liệu nhất định vì tất cả ngữ cảnh mới và từ mới mỗi ngày liên tục xuất hiện đó cũng là vấn đề cho việc gán từ loại thủ công.

## 2. Giới thiệu bài toán

Gán nhãn từ loại là một quá trình xử lý ngôn ngữ tự nhiên (NLP), trong đó các từ trong văn bản được chú thích với danh mục ngữ pháp tương ứng của chúng, chẳng hạn như danh từ, động từ, tính từ, trạng từ,....

Mục tiêu của việc gán thẻ POS là để xác định vai trò của các từ trong câu, điều này rất cần thiết cho các tác vụ như phân tích cú pháp và phân loại văn bản. Gán thẻ POS là một bước cơ bản trong NLP và được sử dụng làm tiền đề cho các bài toán phân loại văn bản, nó có thể áp dụng cho các bài toán phân loại cảm xúc theo chủ đề, phân loại sentiment (cảm xúc),....

Trong đồ án này nhóm sẽ sử dụng mô hình Hidden Markov kết hợp với thuật toán Viterbi để gán nhãn từ loại Tiếng Việt trên bộ ngữ liệu của nhóm tự thu thập và thực hiện so sánh độ chính xác với thư viện VNCORENLP trên bộ ngữ liệu đó.

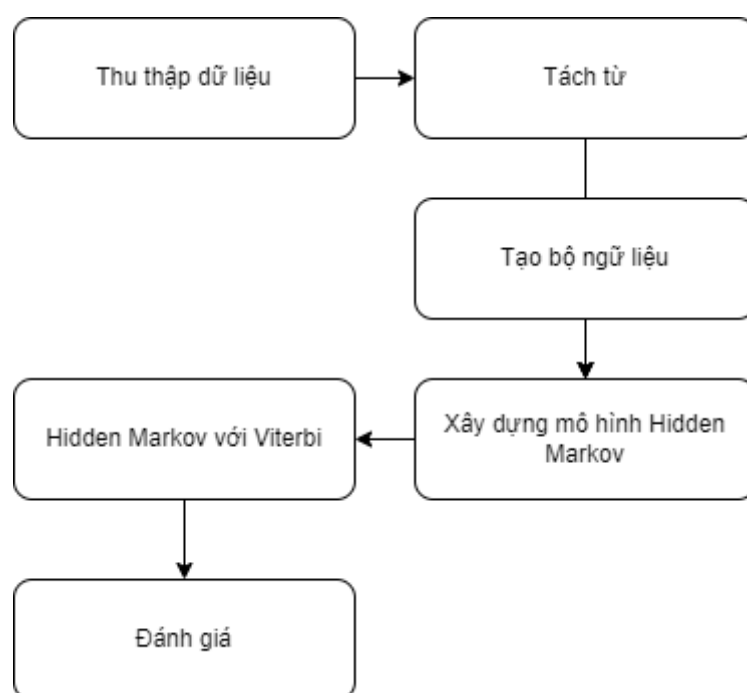
Bài toán gán nhãn từ loại Tiếng Việt.

**Input:** Một câu tiếng Việt bất kỳ.

**Output:** Nhãn của từng từ trong câu.

INPUT	OUTPUT
Cô giáo dạy học sinh học sinh học.	Np, V,N,V,N, CH

Quá trình thực hiện:



## CHƯƠNG II: THU THẬP DỮ LIỆU

### 1. Nguồn thu thập dữ liệu

Nhóm tiến hành thu thập dữ liệu từ mạng xã hội là chính về nhiều lĩnh vực như âm ngôn, tiểu thuyết, báo chí, sách, bài hát, bóng đá... Các thông tin chi tiết bao gồm.

### 2. Bộ dữ liệu

Sau khi tiến hành thu thập nhóm thu được bộ dữ liệu gốc và lưu với tên data\_begin.txt.

- Số lượng câu: 76 câu.

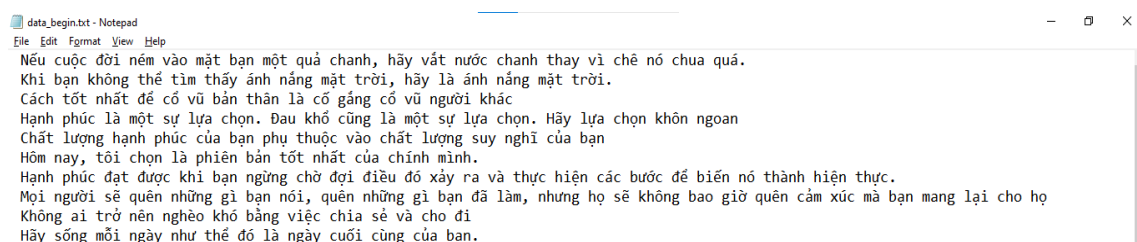
- Mỗi dòng là một câu.
- Các tiếng được phân tách bởi các khoảng trắng.
- Câu dài nhất: 53 tiếng.

*“Các triệu chứng thường thấy là sốt, đau đầu, đau cơ, đau lưng, sưng hạch bạch huyết, ớn lạnh, mệt mỏi, phát ban nhìn giống như mụn nước trên mặt, trong miệng hoặc ở các bộ phận khác của cơ thể như bàn tay, bàn chân, ngực, bộ phận sinh dục, hậu môn.”*

- Câu ngắn nhất: 3 tiếng

*“Bà bó cổ.”*

Tất cả các câu ngữ liệu được lưu vào file `data_begin.txt`.



## CHƯƠNG III: TÁCH TỪ

### 1. Lý do tách từ

Tách từ (word segmentation) là một bước quan trọng trong xử lý ngôn ngữ tự nhiên (NLP). Tách từ cung cấp cho chúng ta định dạng dữ liệu rõ ràng, dễ dàng cho việc xử lý. Khi các từ đã được tách, chúng ta có thể xử lý từng từ riêng biệt, để xác định nghĩa, tính từ vựng hoặc cảm xúc của từ đó. Mục đích của nó giúp chúng ta có thể phân tích một chuỗi từ thành các cụm từ mà máy tính hay thuật toán có thể hiểu và xử lý được.

Ngôn ngữ Tiếng Việt khác với các ngôn ngữ khác, chúng ta không cần phải biến đổi hình thái cho các từ và không xác định ranh giới của các từ bằng các khoảng cách nên một từ có thể chứa nhiều ngữ nghĩa khác nhau, vì vậy việc tách từ là vô cùng quan trọng và là tiền đề để xử lý cho các bài toán lớn hơn.

Trong đề tài này, nhóm dùng ký tự ‘\_’ để biểu thị cho một từ có nhiều hơn một tiếng (từ ghép). Ví dụ:

“Hai con sư tử lựa chọn bắt con thỏ .”

sẽ được tách như sau:

“Hai con sư\_tử lựa\_chọn bắt con thỏ .”

Trong tổng quan, tách từ giúp cho việc xử lý ngôn ngữ tự nhiên trở nên dễ dàng hơn và có độ chính xác cao hơn.

### 3. Tách từ bằng thuật toán Maximum Matching

**Ý tưởng:** Phương pháp này được gọi là so khớp tối đa từ trái sang phải (hoặc ngược lại). Nó sẽ duyệt một câu từ trái sang phải (hoặc ngược lại) và chọn ra từ ghép có độ dài được định nghĩa lớn nhất có mặt trong một từ điển từ vựng được cho sẵn. Quá trình này được lặp đi lặp lại cho đến khi độ dài giảm dần cho đến hết câu.

**Ưu điểm:**

- Thuật toán đơn giản và dễ hiểu.
- Tuy nó sử dụng chiến lược vét cạn nhưng trong thực tế thuật toán này chạy rất nhanh.
- Phù hợp với bộ dữ liệu nhỏ của nhóm.

**Nhược điểm:**

- Nếu các từ không có trong từ điển thì chắc chắn thuật toán sẽ thất bại, không giải quyết được các trường hợp nhập nhằng: có dấu câu, in hoa thường lẫn lộn,...

- Ngoài ra, cách vận hành của thuật toán từ trái sang phải hay phải sang trái cũng có thể cho ra kết quả không nhất quán. Ví dụ: “Học sinh học sinh học” sẽ bị tách như sau “Học\_sinh học\_sinh học” thay vì “Học\_sinh học sinh\_học”.

- Với mục tiêu tách các dấu câu như “,”, “.”, “!”, “?”,... thì Maximum Matching cũng cho thấy nhược điểm của mình khi không thực hiện được. Tuy nhiên nhóm cũng khắc phục điều này khi thêm vào bước tiền xử lý dữ liệu câu input là khi gặp các dấu câu “,”, “.”, “!”, “?”,... thì sẽ có khoảng cách giữa các dấu câu này, như vậy nhược điểm có thể được khắc phục một cách hoàn hảo.

**Ví dụ:** Hà theo mẹ đi thi đấu cờ vua.

**Input:** Hà theo mẹ đi thi đấu cờ vua .

**Output:** Hà theo mẹ đi thi \_đấu cờ \_vua .

Cách triển khai bộ từ điển:

1. Hà theo mẹ đi thi đấu cờ vua.
2. Hai nhà cách nhau một bức tường.
3. Hạnh phúc là một sự lựa chọn
4. Mỗi ngày, sư tử bắt một con thú nhỏ để ăn thịt.

Chúng ta có bộ từ điển gồm 4 câu như sau:

			mỗi
			ngày
		hai	,
			sư_tử
Hà	hạnh_phúc	nhà	bắt
theo	là	cách	một
mẹ	một	nhau	con
đi	sự	một	thú
thi_đấu	lựa_chọn	bức	nhỏ
cờ_vua		tương	để
.	.	.	ăn_thịt
.	.	.	.

Thực hiện tách từ cho câu: Hai con sư tử lựa chọn bắt con thỏ.

Hai con sư tử lựa chọn bắt con thỏ.     Max\_len = 4

Hai con sư tử

Hai con sư

Hai con

Hai

Hai con sư\_tử lựa\_chọn bắt con thỏ.

**Nhược điểm:** không giải quyết được trường hợp dấu câu. Nhóm thực hiện bước tiền xử lý dữ liệu khi thực hiện thêm khoảng cách ngay trước các dấu câu để giải quyết nhược điểm này.

Hai con sư tử lựa chọn bắt con thỏ .

Hai con sư\_tử lựa\_chọn bắt con thỏ .

### 3. Tách từ sử dụng thư viện VnCoreNLP

#### 3.1 Giới thiệu thư viện VnCore NLP

VnCoreNLP là một thư viện tiếng Việt cho xử lý ngôn ngữ tự nhiên nó cung cấp các công cụ và thuật toán cho việc xử lý ngôn ngữ tự nhiên, bao gồm phân tách cấu trúc câu, phân tách từ, từ loại và gán nhãn. VnCoreNLP cung cấp một số tính năng mạnh mẽ và dễ sử dụng cho những nghiên cứu và ứng dụng NLP trên tiếng Việt.

#### 3.2 Cài đặt

Nhóm sử dụng công cụ Google Colab để tiến hành cài đặt thư viện VnCoreNLP, sau khi cài đặt sử dụng phương thức “word\_segment” của thư viện VnCoreNLP để tiến hành tách từ trên bộ dữ liệu đã thu thập.



data\_begin - Notepad  
File Edit Format View Help  
Nếu cuộc đời ném vào mặt bạn một quả chanh, hãy vắt nước chanh thay vì chê nó chua quá.  
Khi bạn không thể tìm thấy ánh nắng mặt trời, hãy là ánh nắng mặt trời.  
Cách tốt nhất để cố vũ bản thân là cố gắng cố vũ người khác  
Hạnh phúc là một sự lựa chọn. Đau khổ cũng là một sự lựa chọn. Hãy lựa chọn khôn ngoan  
Chất lượng hạnh phúc của bạn phụ thuộc vào chất lượng suy nghĩ của bạn  
Hôm nay, tôi chọn là phiên bản tốt nhất của chính mình.



File Edit Format View Help	File Edit Format View Help	File Edit Format View Help
Nếu	Hạnh_phúc	.
cuộc_đời	là	Hạnh_phúc
ném	một	đạt
vào	sự	được
mặt	lựa_chọn	khi
bạn	.	bạn
một	Chất_lượng	ngừng
quả	hạnh_phúc	chờ_đợi
chanh	của	điều
,	bạn	đó
hãy	phụ_thuộc	xảy
vắt	vào	ra
nước	chất_lượng	và
chanh	suy_nghĩ	thực_hiện
thay_vì	của	các
chê	bạn	bước
nó	Hôm_nay	để
chua	,	biến
quá	tôi	nó
.	chọn	thành
Khi	là	hiện_thực
bạn	phiên_bản	.
không_thể	tốt	Mọi
tìm	nhất	người
thấy	của	sẽ
ánh	chính	quên
nắng		những
mặt_trời		.

### Ưu điểm:

- Tốc độ xử lý nhanh: VnCoreNLP xử lý tốt dữ liệu tiếng Việt với tốc độ khá nhanh.
- Dễ sử dụng: là một thư viện được xây dựng sẵn nên dễ dàng cài đặt và sử dụng
- Chất lượng tách từ tốt: VnCoreNLP cung cấp một kết quả tách từ khá tốt, giúp cho việc xử lý tiếng Việt dễ dàng hơn.

### Nhược điểm:

- Có thể có một số lỗi sai về các từ mới hoặc từ lạ, ví dụ trong trường hợp bộ dữ
- VD: “*Tập đoàn của ông Alshaya sở hữu nhượng quyền thương mại địa phương của các thương hiệu bán lẻ như Starbucks, H&M và Victoria’s Secret.*”

,	Tập_đoàn	Một
ngụ	của	số
Tuyên_Quang	ông	vùng
,	Alshaya	ven
du_lịch	sở_hữu	biển
từ	nhượng	của
Dubai	quyền	Thừa_Thiên_Huế
về	thương_mại	ngập_úng
sân_bay	địa_phương	gần
Tân_Sơn_Nhất	của	một
triệu_chứng	các	tuần
sốt	thương_hiệu	nay
,	bán_lẻ	.
được	như	
cách_ly	Starbucks	
ngay	,	
tại	H	
sân_bay	&	
và	M	
lấy	và	
mẫu	Victoria	
xét_nghiệm	,	
,	s	
mắc	Secret	
đậu_mùa_khí	.	

Trong ví dụ trên các tên riêng “H&M” hay “Victoria’s Secret”, “Tân Sơn Nhất”, “Thừa Thiên Huế” thư viện VNCORE nó sẽ bị sai sót khi không thể tách đúng theo tên riêng.

,	
H	Victoria
&	,
M	s
và	Secret
Victoria	.
,	

## CHƯƠNG IV: TẠO BỘ NGỮ LIỆU

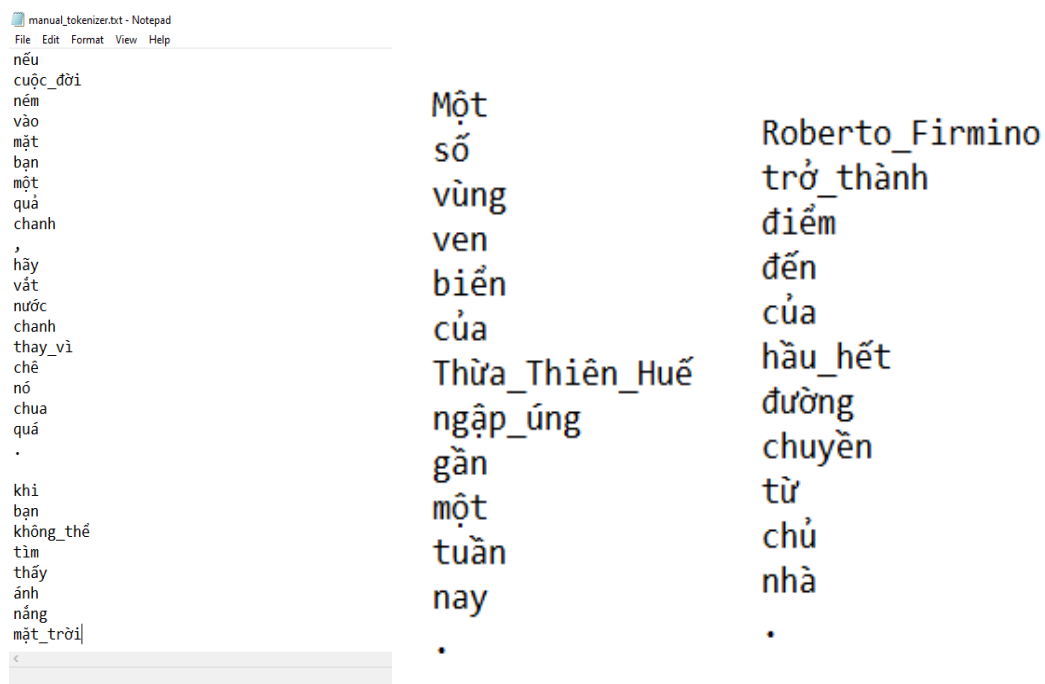
### 1. Tạo bộ ngữ liệu

#### 1.1. Tách từ thủ công

Nhóm thực hiện tách từ thủ công bằng cách tra từ điển VLSP tại trang sau: <https://vlsp.hpda.vn/demo/?page=vcl>.

Bộ dữ liệu được tách và lưu tại file `manual_tokenizer.txt` với các yêu cầu sau:

- Thực hiện chèn ký hiệu ‘\_’ vào giữa các từ ghép, mỗi dòng là 1 từ
- Kết thúc câu bằng ký tự xuống dòng ‘\n’
- Số lượng từ: **1168 từ**
- Số lượng từ ghép: **221 từ**



**Trường hợp nhập nhằng:** Khi thực hiện tra từ điển VLSP với mục tiêu tách từ, nhóm nhận ra một số từ trong ngữ liệu thu thập có các trường hợp nhập nhằng như: khi nào, chỉ cần, cùng, thế nào, sau đó, đã, không thể, đi chợ, thi hẹp, vào, tìm thấy, người ta, những người, qua.....

Sau khi đã thực hiện tách từ thủ công, dữ liệu sẽ mang ý nghĩa là ground-truth để làm cơ sở so sánh các phương pháp tách từ tự động. 2 phương pháp tách từ tự động nhóm em sử dụng là: Maximum Matching và sử dụng thư viện VnCoreNLP. Trong đó, thuật toán Maximum Matching nhóm sử dụng với độ dài định nghĩa **maxlen=4**. Kết quả được đánh giá cụ thể như sau:

	Maximum Matching	VnCoreNLP
Accuracy	83%	79%

Dựa vào bảng kết quả của việc đánh giá kết quả tách từ trên bộ dữ liệu của nhóm, có thể nhận thấy accuracy của phương pháp Maximum Matching cho độ chính xác cao hơn so với phương pháp tách từ bằng thư viện VnCoreNLP. Nguyên nhân chủ yếu do bộ dữ liệu chúng em thu thập và sử dụng có nhiều danh từ riêng, phương pháp VnCoreNLP cho kết quả không đúng. Ví dụ “H&M”, “Victoria’s Secret”,... Phương pháp Maximum Matching cho kết quả đúng tuy nhiên VnCoreNLP thì cho kết quả chưa chính xác.

## 1.2. Quy trình tạo ngữ liệu và gán nhãn

### 1.2.1. Gán nhãn thủ công

Sau khi có được kết quả tách từ thủ công nhóm thực hiện gán nhãn thủ công trên kết quả tách từ thủ công. Bằng cách tra tất cả các từ trong file tách từ thu được trên từ điển VLSP tại <https://vlsp.hpda.vn/demo/?page=vcl> nhóm đã thống nhất các trường hợp nhập những từ loại. Quy trình gán nhãn được thực hiện tuần tự theo với từng câu trong bộ ngữ liệu. Trong đó, khoảng trắng xuống hàng “\n” không cần gán nhãn, bởi vì nó biểu thị cho vị trí kết thúc của một câu. Nhãn của các từ được gán theo quy tắc:

# ĐỀ TÀI VLSP

## Nhánh đề tài XỬ LÝ VĂN BẢN

[Trang chủ](#)[Từ điển](#)[Phân tích từ/cụm từ](#)[Phân tích cú pháp](#)[Tài nguyên](#)[Giới thiệu](#)

### Vietnamese Computational Lexicon

[Nhãn từ loại](#) | [Khung vị từ](#) | [Cây ngữ nghĩa](#) | [Vai nghĩa](#)

Nhập từ cần tra cứu:

hạnh phúc

Tra từ

Từ **hạnh phúc** có 2 nghĩa.

XEM TẤT CẢ

1. **hạnh phúc** (N) trạng thái sung sướng vì cảm thấy hoàn toàn đạt được ý nguyện

2. **hạnh phúc** (A) có được trạng thái sung sướng vì cảm thấy đã đạt được ý nguyện

Bộ ngữ liệu sau khi gán nhãn thủ công dựa theo thư viện VLSP được lưu tại file manual\_pos\_tag.txt.

manual_pos_tag.txt - Notepad	manual_pos_tag.txt - Notepad	manual_pos_tag.txt - Notepad
File Edit Format View Help	File Edit Format View Help	File Edit Format View Help
Nếu C	Cách N	Hôm nay N
cuộc đời N	tốt A	, CH
nắm V	nhất R	tôi P
vào E	để C	chọn V
mặt N	cổ vũ V	là I
bạn N	bản thân P	phiên bản N
một M	là C	tốt A
quả N	cố gắng V	nhất R
chanh N	cổ vũ V	của E
, CH	người N	chính A
hãy R	khác A	mình P
vắt V	. CH	. CH
nước N	Hạnh phúc N	Hạnh phúc N
chanh N	là C	đạt V
thay vì X	một M	được A
chê V	sự N	khi N
nó P	lựa chọn V	bạn P
chưa A	. CH	ngừng V
quả R	Chất lượng N	chờ đợi V
. CH	hạnh phúc N	điều N
Khi N	của E	đó P
bạn N	bạn N	xây V
không thể X	phụ thuộc V	ra V
tìm V	vào E	và Cc
thấy V	chất lượng N	thực hiện V
ánh N	suy nghĩ V	các L
nắng N		bước N
mặt trời N		để C

### 1.2.2. Gán nhãn bằng thư viện VNCORE NLP

Ngoài việc gán nhãn thủ công bằng cách sử dụng thư viện VLSP. Nhóm còn tiến hành gán nhãn cho ngữ liệu bằng thư viện VNCORENLP. Bộ ngữ liệu sau khi sử dụng thư viện VNCORE NLP để gán nhãn được lưu vào file với tên là `vncore_pos_tag.txt`

File	Edit	View
Nếu C		
cuộc_đời N		
ném V		
vào E		
mặt N		
bạn N		
một M		
quả N		
chanh V		
, CH		
hãy R		
vắt V		
nước N		
chanh N		
thay_vì X		
chê M		
nó P		
chua A		
quá R		
. CH		
Khi N		
bạn N		
không_thể R		
tìm V		
.. ~		

File	Edit	View
Cách V		
tốt A		
nhất A		
để E		
cổ_vũ V		
bản_thân N		
là V		
cổ_gắng V		
cổ_vũ V		
người N		
khác A		
Hạnh_phúc N		
là V		
một M		
sự N		
lựa_chọn V		
. CH		
Chất_lượng N		
hạnh_phúc N		
của E		
bạn N		
phụ_thuộc V		
vào E		
chất_lượng N		

File	Edit	View
Hôm_nay N		
, CH		
tôi P		
chọn V		
là V		
phiên_bản N		
tốt A		
nhất A		
của E		
chính T		
mình P		
. CH		
Hạnh_phúc N		
đạt V		
được V		
khi N		
bạn N		
ngừng V		
chờ_đợi V		
điều N		
đó P		
xảy V		
ra V		
và Cc		
thực_hiện V		

### 1.2.3. Tạo bộ ngữ liệu

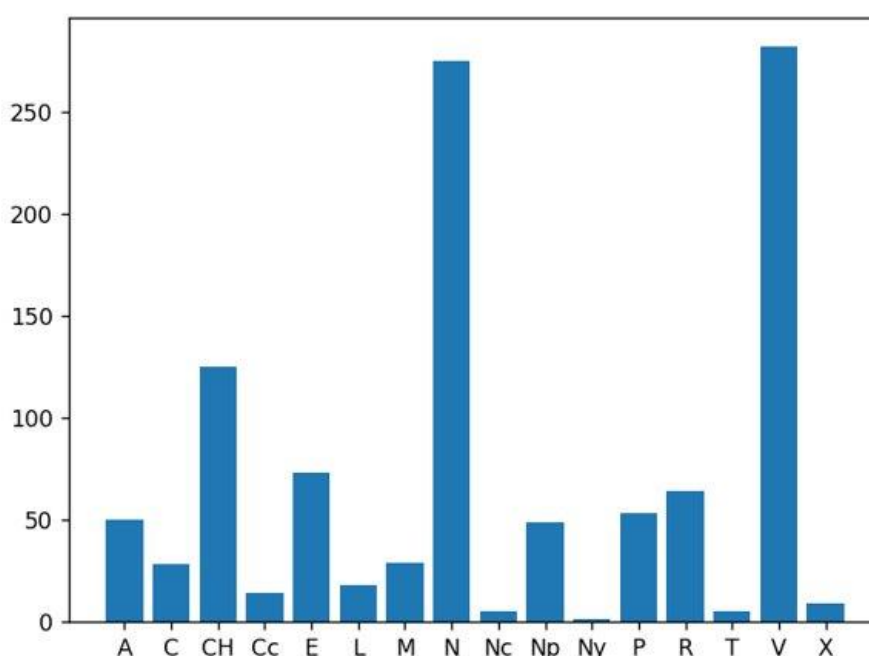
Có được bộ ngữ liệu gán nhãn, nhóm tiến hành chia 76 câu ngữ liệu thành 2 phần gồm tập Train và tập Test với số lượng như sau: 62 câu cho tập Train, 14 câu cho tập test.

Các file liên quan đến ngữ liệu gán nhãn được lưu trong thư mục **data\_tag**. Trong đó các file có ý nghĩa như sau:

- Một file `vncore_pos_tag.txt` chứa ngữ liệu là 76 câu đã được nhóm gán nhãn thủ công, mỗi từ và nhãn ứng với một hàng, các từ và nhãn cách nhau bằng một phím tab “\t”.
- Một file `manual_pos_tag.txt` chứa ngữ liệu là 76 câu đã được nhóm gán nhãn thủ công, mỗi từ và nhãn ứng với một hàng, các từ và nhãn cách nhau bằng một phím tab “\t”.

- Một file data\_train.txt có cấu trúc tương tự với file gán nhãn thủ bằng thư viện VnCoreNLP, chứa các từ và nhãn để sử dụng cho việc huấn luyện mô hình Hidden Markov.
- Một file data\_train\_notag.txt chứa dữ liệu gồm các từ đã tách bằng thư viện VnCoreNLP không có nhãn để sử dụng cho việc đánh giá mô hình trên tập train.
- Một file data\_test chứa dữ liệu chưa gán nhãn bằng thư viện VnCoreNLP sử dụng để đánh giá độ chính xác của mô hình trên tập dữ liệu không được mô hình nhìn thấy.

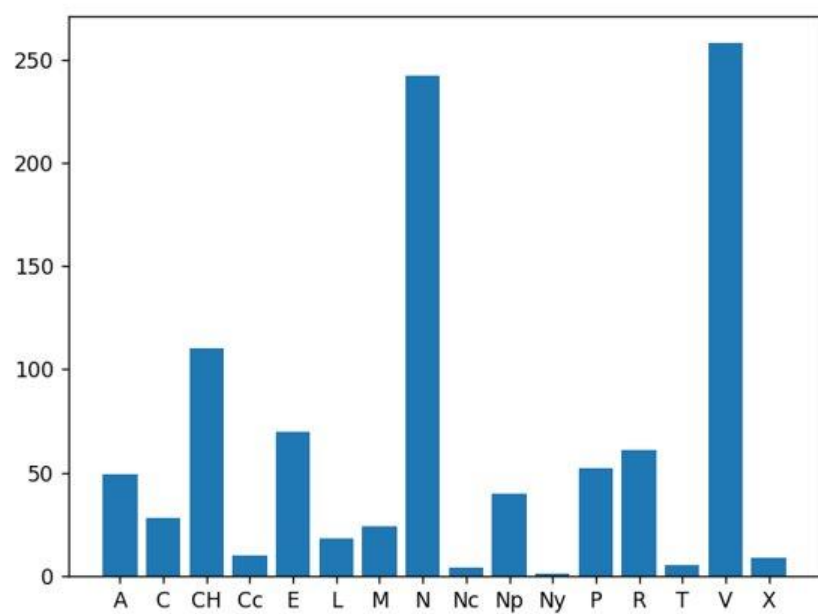
Bộ ngữ liệu gồm 76 câu:



Toàn bộ ngữ liệu và phân bố nhãn của bộ ngữ liệu gán bằng VnCoreNLP:

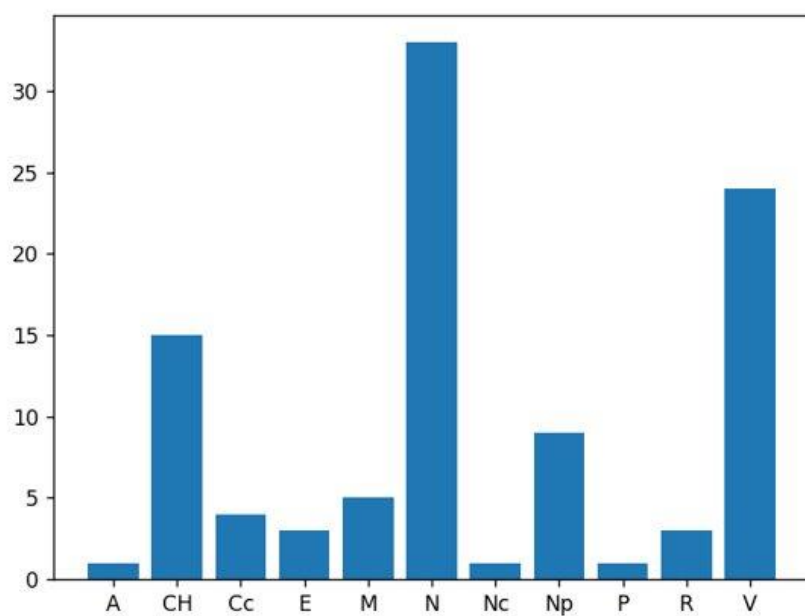
CH	X	N	T	V	M	A	L	E	Cc	Np	P	R	C	Nc	Ny
125	9	275	5	282	29	50	18	73	14	49	53	64	28	5	1

Tập Train: 62 câu



CH	X	N	T	V	M	A	L	E	Cc	Np	P	R	C	Nc	Ny
110	9	242	5	258	24	49	18	70	10	40	52	61	28	4	1

Tập Test: 14 câu



CH	N	V	M	A	E	Cc	Np	P	R	Nc
15	33	24	5	1	3	4	9	1	3	1



# CHƯƠNG V: XÂY DỰNG MÔ HÌNH HIDDEN MARKOV

## 1. Hidden Markov

### 1.1. Giới thiệu Hidden Markov

Mô hình Hidden Markov (Hidden Markov Model - HMM) là một mô hình thống kê đại diện cho một quá trình Markov với các trạng thái không quan sát được (hidden states) và các trạng thái quan sát được (observed states). Trong bài toán gán nhãn dữ liệu trạng thái quan sát được là các từ như là đầu vào của mô hình, trạng thái ẩn là các nhãn. HMM có thể được sử dụng không chỉ để gán nhãn từ loại mà còn để nhận dạng giọng nói, tổng hợp giọng nói và nhiều ứng dụng khác.

Mô hình Markov chứa một số trạng thái và xác suất chuyển đổi giữa các trạng thái đó. Mô hình Markov sử dụng ma trận chuyển tiếp A (Transition Matrix). Mô hình Markov ẩn thêm một ma trận phát xạ B (Emission Matrix) mô tả xác suất của một quan sát có thể nhìn thấy khi ta ở một trạng thái cụ thể.

Bài toán trong HMM mà nhóm chúng em giải quyết là bài toán học dựa trên dữ liệu đã được gán nhãn. Để có thể đánh giá được độ chính xác của mô hình nhóm thực hiện 2 so sánh đó là so sánh sử dụng Hidden Markov trên 2 bộ ngữ liệu tách tay và bộ ngữ liệu được tách bằng thư viện VNCORENLP và so sánh giữa Hidden Markov sử dụng bộ ngữ liệu VNCORENLP với thư viện VNCORENLP sử dụng bộ ngữ liệu được tách bằng VNCORENLP.

**Input:** Một câu

**Output:** Nhãn của từng từ trong câu

### 1.2. Ma trận chuyển trạng thái A

Ma trận trạng thái là một ma trận xác định các xác suất chuyển đổi từ một trạng thái đến các trạng thái khác trong một quá trình Markov.

**Cách tính ma trận A:**

Bước 1: Thống kê tần số. Ở đây ta thống kê xem từ trạng thái này đến trạng thái khác trong ngữ liệu có bao nhiêu trường hợp.

Vd: Như trường hợp trên từ nhãn E đến nhãn E là 1 trường hợp.

	E	M	V	T	Ny	Nc	...	L	X	C	R	Np	Cc
<s>	4.0	2.0	6.0	0.0	0.0	0.0	...	2.0	0.0	3.0	5.0	19.0	0.0
E	1.0	2.0	8.0	1.0	0.0	1.0	...	4.0	1.0	1.0	1.0	3.0	0.0
M	1.0	0.0	5.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
V	24.0	4.0	70.0	1.0	0.0	2.0	...	8.0	1.0	5.0	10.0	4.0	4.0
T	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0
Ny	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0
Nc	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
A	9.0	3.0	4.0	0.0	0.0	0.0	...	0.0	0.0	4.0	2.0	0.0	2.0
P	1.0	0.0	18.0	0.0	0.0	0.0	...	2.0	0.0	2.0	10.0	0.0	0.0
N	13.0	10.0	58.0	3.0	0.0	0.0	...	0.0	3.0	8.0	14.0	7.0	3.0
CH	3.0	0.0	22.0	0.0	1.0	0.0	...	1.0	1.0	3.0	5.0	2.0	0.0
L	0.0	0.0	2.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0
X	2.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
C	1.0	2.0	7.0	0.0	0.0	0.0	...	1.0	1.0	0.0	3.0	1.0	0.0
R	3.0	0.0	36.0	0.0	0.0	0.0	...	0.0	0.0	1.0	6.0	0.0	0.0
Np	5.0	0.0	11.0	0.0	0.0	0.0	...	0.0	1.0	0.0	2.0	3.0	0.0
Cc	2.0	0.0	5.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	1.0	0.0

Bước 2: Điều chỉnh tần số. Điều này giúp tăng độ chính xác của mô hình, nó giúp mô hình thể hiện được những chuyển đổi trạng thái có thể có mà không thống kê được trong tập ngữ liệu. Ở trong bài toán của chúng em chúng em sử dụng Laplace + 1.

	E	M	V	T	Ny	Nc	...	L	X	C	R	Np	Cc
<s>	5.0	3.0	7.0	1.0	1.0	1.0	...	3.0	1.0	4.0	6.0	20.0	1.0
E	2.0	3.0	9.0	2.0	1.0	2.0	...	5.0	2.0	2.0	2.0	4.0	1.0
M	2.0	1.0	6.0	1.0	1.0	1.0	...	1.0	1.0	1.0	1.0	1.0	1.0
V	25.0	5.0	71.0	2.0	1.0	3.0	...	9.0	2.0	6.0	11.0	5.0	5.0
T	2.0	1.0	1.0	1.0	1.0	1.0	...	1.0	1.0	2.0	1.0	1.0	1.0
Ny	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	1.0	1.0	1.0	1.0	2.0
Nc	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	1.0	1.0	1.0	1.0	1.0
A	10.0	4.0	5.0	1.0	1.0	1.0	...	1.0	1.0	5.0	3.0	1.0	3.0
P	2.0	1.0	19.0	1.0	1.0	1.0	...	3.0	1.0	3.0	11.0	1.0	1.0
N	14.0	11.0	59.0	4.0	1.0	1.0	...	1.0	4.0	9.0	15.0	8.0	4.0
CH	4.0	1.0	23.0	1.0	2.0	1.0	...	2.0	2.0	4.0	6.0	3.0	1.0
L	1.0	1.0	3.0	1.0	1.0	2.0	...	1.0	1.0	1.0	1.0	1.0	1.0
X	3.0	2.0	2.0	1.0	1.0	1.0	...	1.0	1.0	1.0	1.0	1.0	1.0
C	2.0	3.0	8.0	1.0	1.0	1.0	...	2.0	2.0	1.0	4.0	2.0	1.0
R	4.0	1.0	37.0	1.0	1.0	1.0	...	1.0	1.0	2.0	7.0	1.0	1.0
Np	6.0	1.0	12.0	1.0	1.0	1.0	...	1.0	2.0	1.0	3.0	4.0	1.0
Cc	3.0	1.0	6.0	1.0	1.0	1.0	...	1.0	1.0	1.0	2.0	2.0	1.0

Bước 3: Tính xác suất smoothing: Cuối cùng là tính xác suất xem từ trạng thái này sang trạng thái khác bằng có tỉ lệ là bao nhiêu. Xác suất của mỗi ô được tính bằng giá trị trong ô chia cho tổng giá trị của hàng chứa ô đó.

	E	M	V	...	R	Np	Cc
<s>	0.064935	0.038961	0.090909	...	0.077922	0.259740	0.012987
E	0.023256	0.034884	0.104651	...	0.023256	0.046512	0.011628
M	0.050000	0.025000	0.150000	...	0.025000	0.025000	0.025000
V	0.093284	0.018657	0.264925	...	0.041045	0.018657	0.018657
T	0.095238	0.047619	0.047619	...	0.047619	0.047619	0.047619
Ny	0.058824	0.058824	0.058824	...	0.058824	0.058824	0.117647
Nc	0.050000	0.050000	0.050000	...	0.050000	0.050000	0.050000
A	0.156250	0.062500	0.078125	...	0.046875	0.015625	0.046875
P	0.030769	0.015385	0.292308	...	0.169231	0.015385	0.015385
N	0.054902	0.043137	0.231373	...	0.058824	0.031373	0.015686
CH	0.058824	0.014706	0.338235	...	0.088235	0.044118	0.014706
L	0.029412	0.029412	0.088235	...	0.029412	0.029412	0.029412
X	0.125000	0.083333	0.083333	...	0.041667	0.041667	0.041667
C	0.045455	0.068182	0.181818	...	0.090909	0.045455	0.022727
R	0.053333	0.013333	0.493333	...	0.093333	0.013333	0.013333
Np	0.107143	0.017857	0.214286	...	0.053571	0.071429	0.017857
Cc	0.115385	0.038462	0.230769	...	0.076923	0.076923	0.038462

Hình 3

### 1.3. Ma trận thể hiện B (Emission Matrix)

Ma trận thể hiện B cho biết xác suất một trạng thái quan sát với một nhãn

Ma trận B có kích thước (N\*M) trong đó:

N là số nhãn

M là số từ

Cách tính ma trận B: Tính ma trận B có tác tính khá tương tự với cách tính ma trận A.

Bước 1: Thống kê tần số. Tại đây ta sẽ thống kê với 1 từ thì các nhãn tương ứng với từ đó xuất hiện bao nhiêu lần.

	tại	xăng dầu	sinh dục	mọi	...	lãnh đạo	Quân Vương	ăn miếng	trà miếng	hay
P	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
CH	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
M	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
A	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
N	0.0	1.0	0.0	0.0	...	1.0	2.0		0.0	0.0
X	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
T	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
Cc	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
Ny	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
Nc	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
R	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
Np	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
L	0.0	0.0	0.0	1.0	...	0.0	0.0		0.0	0.0
V	0.0	0.0	1.0	0.0	...	0.0	0.0		1.0	0.0
E	1.0	0.0	0.0	0.0	...	0.0	0.0		0.0	0.0
C	0.0	0.0	0.0	0.0	...	0.0	0.0		0.0	1.0

Bước 2: Điều chỉnh tần số. Bước này tương tự với cách tính ma trận A.

[16 rows x 546 columns]										
	tại	xăng dầu	sinh dục	mọi	...	lãnh đạo	Quân Vương	ăn miếng	trà miếng	hay
P	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
CH	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
M	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
A	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
N	1.0	2.0	1.0	1.0	...	2.0	3.0		1.0	1.0
X	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
T	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
Cc	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
Ny	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
Nc	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
R	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
Np	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
L	1.0	1.0	1.0	2.0	...	1.0	1.0		1.0	1.0
V	1.0	1.0	2.0	1.0	...	1.0	1.0		2.0	1.0
E	2.0	1.0	1.0	1.0	...	1.0	1.0		1.0	1.0
C	1.0	1.0	1.0	1.0	...	1.0	1.0		1.0	2.0

Hình 4

Bước 3: Tính xác suất. Bước này cũng có cách tính tương tự với cách tính của bước 3 của ma trận A.

	tại	xăng dầu	sinh dục	...	Quân Vương	ăn miếng	trà miếng	hay
P	0.001678	0.001678	0.001678	...	0.001678		0.001678	0.001678
CH	0.001527	0.001527	0.001527	...	0.001527		0.001527	0.001527
M	0.001754	0.001754	0.001754	...	0.001754		0.001754	0.001754
A	0.001681	0.001681	0.001681	...	0.001681		0.001681	0.001681
N	0.001272	0.002545	0.001272	...	0.003817		0.001272	0.001272
X	0.001805	0.001805	0.001805	...	0.001805		0.001805	0.001805
T	0.001815	0.001815	0.001815	...	0.001815		0.001815	0.001815
Cc	0.001799	0.001799	0.001799	...	0.001799		0.001799	0.001799
Ny	0.001828	0.001828	0.001828	...	0.001828		0.001828	0.001828
Nc	0.001818	0.001818	0.001818	...	0.001818		0.001818	0.001818
R	0.001653	0.001653	0.001653	...	0.001653		0.001653	0.001653
Np	0.001706	0.001706	0.001706	...	0.001706		0.001706	0.001706
L	0.001773	0.001773	0.001773	...	0.001773		0.001773	0.001773
V	0.001252	0.001252	0.002503	...	0.001252		0.002503	0.001252
E	0.003247	0.001623	0.001623	...	0.001623		0.001623	0.001623
C	0.001742	0.001742	0.001742	...	0.001742		0.001742	0.003484

## 1.4. Gán nhãn:

Cho X là chuỗi trạng thái ẩn và O là chuỗi trạng thái quan sát được, khi đó

$$P(X, O|M) = \prod_{t=1}^N A[x_{t-1}, x_t]B(x_t, o_t)$$

$$P(O|M) = \sum_x p(X, O|M)$$

Tìm X sao cho xác suất  $P(X, O|M)$  lớn nhất.

Sau khi tính được ma trận A thì chúng em thực hiện bước gán nhãn cho chuỗi ở đây chúng em gặp hai vấn đề:

- + Nếu sử dụng Brute Force để duyệt toàn bộ trường hợp thì số trường hợp có khả năng xuất hiện quá nhiều nên không khả thi.
- + Từ mà chúng em muốn gán nhãn là một từ không được thống kê trong ma trận thể hiện B.

### Cách giải quyết

Sử dụng thuật toán Viterbi

```

function VITERBI(observations of len  $T$ , state-graph of len  $N$ ) returns best-path, path-prob

create a path probability matrix viterbi[ $N, T$ ]
for each state  $s$  from 1 to  $N$  do                                ; initialization step
    viterbi[ $s, 1$ ]  $\leftarrow \pi_s * b_s(o_1)$ 
    backpointer[ $s, 1$ ]  $\leftarrow 0$ 
for each time step  $t$  from 2 to  $T$  do                                ; recursion step
    for each state  $s$  from 1 to  $N$  do
        viterbi[ $s, t$ ]  $\leftarrow \max_{s'=1}^N \text{viterbi}[s', t-1] * a_{s',s} * b_s(o_t)$ 
        backpointer[ $s, t$ ]  $\leftarrow \operatorname{argmax}_{s'=1}^N \text{viterbi}[s', t-1] * a_{s',s} * b_s(o_t)$ 
    bestpathprob  $\leftarrow \max_{s=1}^N \text{viterbi}[s, T]$                                 ; termination step
    bestpathpointer  $\leftarrow \operatorname{argmax}_{s=1}^N \text{viterbi}[s, T]$                                 ; termination step
    bestpath  $\leftarrow$  the path starting at state bestpathpointer, that follows backpointer[] to states back in time
return bestpath, bestpathprob

```

Ý tưởng chính của thuật toán Viterbi là tính toán độ tin cậy của một chuỗi trạng thái dựa trên xác suất tối đa của các chuỗi trạng thái có thể dẫn đến chuỗi quan sát. Cụ thể, thuật toán sử dụng một ma trận trạng thái (state matrix) để lưu trữ các giá trị độ tin cậy tại mỗi trạng thái cho mỗi vị trí trong chuỗi quan sát. Bắt đầu với trạng thái ban đầu, thuật toán tìm kiếm độ tin cậy tại mỗi trạng thái kế tiếp bằng cách tìm kiếm giá trị lớn nhất của tích giữa xác suất chuyển trạng thái (transition probability) từ trạng thái hiện tại đến trạng thái kế tiếp và xác suất phát sinh quan sát (emission probability) tại trạng thái kế tiếp. Sau đó, thuật toán lưu trữ giá trị tìm được vào ma trận trạng thái và di chuyển tới trạng thái kế tiếp để tiếp tục quá trình tính toán.

Trong quá trình tính toán xác suất sẽ xuất hiện trường hợp từ không xuất hiện trong ma trận thể hiện B. Cách giải quyết của chúng em là đặt giá trị thể hiện các nhãn của từ đó là 1 sau đó tiếp tục tính toán như bình thường.

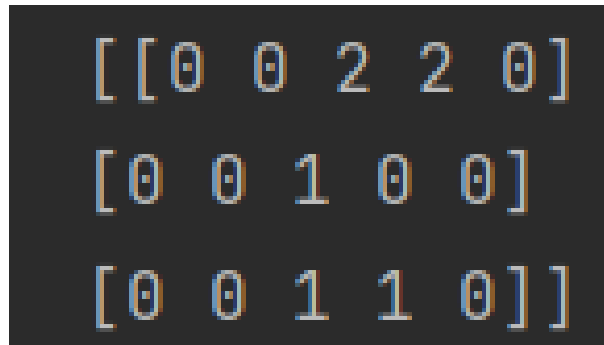
VD: Kết quả của thuật toán Viterbi.

Ma trận trạng thái.

	con	trèo	là	con	nào
UN	0.256410	0.002192	0.005698	0.002622	0.000291
NN	0.012821	0.015341	0.002557	0.000341	0.002040
VB	0.016667	0.008547	0.010227	0.000170	0.000291

Hình 5

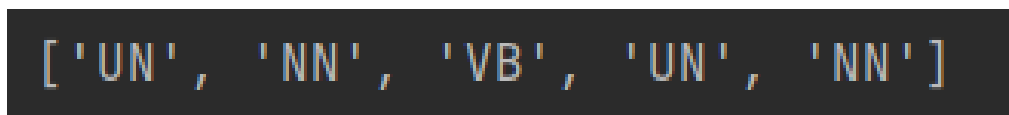
Ma trận lưu lại vị trí trước đó.



```
[[0 0 2 2 0]
 [0 0 1 0 0]
 [0 0 1 1 0]]
```

Hình 6

Kết quả:



```
['UN', 'NN', 'VB', 'UN', 'NN']
```

Hình 7

Như ta có thể thấy hình 8 thể hiện xác suất tốt nhất của chuỗi. Còn hình 9 thể hiện vị trí của nhãn trước đó tạo thành xác suất đó. Ta tìm nhãn theo cách:

Từ	Nhãn	Vị trí trước đó
nào	NN	0
con	UN	2
là	VB	1
trèo	NN	0
con	UN	

Bảng 1: Thể hiện cách tìm nhãn của thuật toán Viterbi



# CHƯƠNG VI: ĐÁNH GIÁ

## 1. Cách tính accuracy

$$\text{Acc} = \frac{\text{Số từ đúng}}{\text{Tổng số từ}}$$

Phương pháp	Accuracy
HMM/Bộ ngữ liệu VNCoreNlp(train)	61,3%
HMM/Bộ ngữ liệu VNCoreNlp(test)	60,6%
HMM/Bộ ngữ liệu tách tay(train)	60,7%
HMM/Bộ ngữ liệu tách tay(test)	68,4%
VnCoreNlp(train)	82%
VnCoreNlp(test)	83%

Nhận xét:

- Trên cùng một mô hình huấn luyện bộ ngữ liệu được tách bằng tay có độ chính xác cao hơn ở tập test vì do số lượng nhãn ở trong bộ ngữ liệu được tách bằng tay ít hơn số lượng nhãn của bộ ngữ liệu được tách bằng thư viện VnCore Nlp trong khi đó số câu lại bằng nhau.
- Với cùng một bộ ngữ liệu là VNCoreNlp thì gán nhãn bằng thư viện VNCoreNlp có độ chính xác lớn hơn do thư viện này được công bố rộng rãi đã qua sự điều chỉnh của các chuyên gia. Còn mô hình Hidden Markov của chúng em có độ chính xác thấp do sự thiếu kinh nghiệm và xử lý các trường hợp chưa tối ưu.

## 2. Kết luận

- Dữ liệu quá ít không bao quát hết ngữ cảnh, khả năng xử lý từ mới không có trong ma trận thể hiện chưa thực sự tối ưu.
- Hướng phát triển:
  - + Bổ sung bộ dữ liệu.
  - + Cải thiện phương pháp gán nhãn.



# CHƯƠNG VII: TÀI LIỆU THAM KHẢO

- [1] . Từ điển VLSP, URL <https://vlsp.hpda.vn/demo/?page=vcl>
- [2] . Github repo DS4V, URL <https://github.com/ds4v/vietnamese-pos-tagging>
- [3] . Github repo buidung2004, URL: <https://github.com/buidung2004/POS-Tagging-Vietnamese>
- [4] . Báo VNExpress, URL: <https://vnexpress.net/>
- [5] . Báo ZingNews, URL: <https://zingnews.vn/>
- [6] . Stanford, URL: <https://web.stanford.edu/~jurafsky/slp3/>