ĐẠI HỌC QUỐC GIA TP. HCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO TỔNG KẾT

ĐỀ TÀI XÂY DỰNG MÔ HÌNH HIDDEN MARKOV ĐỂ GÁN NHÃN CHO TỪ LOẠI TIẾNG VIỆT

Khoa: Khoa học máy tính

Môn: Xử lý ngôn ngữ tự nhiên – CS221.M11

Giáo viên hướng dẫn: ThS. Nguyễn Trọng Chỉnh

Tham gia thực hiện

TT	Họ và tên, MSSV	Chịu trách nhiệm	Điện thoại	Email
1.	Bùi Trần Ngọc Dũng	Leader	0762632004	19521385@gm.uit.edu.vn
2.	Nguyễn Đăng Minh	Member		19520164@gm.uit.edu.vn

Thành phố Hồ Chí Minh – Tháng 12 /2021

BẢNG PHÂN CÔNG

Công việc	Phân công	Tỉ lệ hoàn thành
Thu thập ngữ liệu	Dũng	100%
Xử lý tách từ	Dũng, Minh	60% - 40%
Gán nhãn thủ công	Dũng, Minh	50% - 50%
Huấn luyện mô hình và đánh giá	Dũng	60% - 40%
Viết báo cáo	Dũng, Minh	70% - 30%

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

HMM	Hidden Markov Model
VLSP	Vietnamese Language and Speech Processing
CRF	Conditional Random Fields

Chương 1. TỔNG QUAN

1.1. Giới thiệu chung

Xã hội càng phát triển cùng với nền văn minh văn hóa nhân loại và khoa học kỹ thuật. Máy tính thông minh ra đời đánh dấu cho một bước nhảy vọt, giúp ích cho con người rất nhiều trong mọi lĩnh vực. Tuy nhiên, với mỗi nền văn hóa khác nhau, những ngôn ngữ khác nhau để giúp cho máy tính có thể hiểu được hoàn toàn ngôn ngữ tự nhiên như con người là không hề dễ. Đó chính là mục đích ra đời của môn học xử lý ngôn ngữ tự nhiên

Trong Xử lý ngôn ngữ tự nhiên, có rất nhiều vấn đề đáng lưu tâm: Thực hiện tách từ để xác định các từ đơn ghép, giúp cho tránh các tình trạng nhập nhằng khó cắt nghĩa của ngôn ngữ, hay là việc phân tích từ pháp cho các từ loại từ đó xác định rõ ràng từ loại trong câu giúp giải quyết các vấn đề kết hợp từ trong câu, hoặc phân tích cú pháp để máy tính có thể hiểu cấu trúc của câu,….

Trong đồ án này, nhóm em sẽ ứng dụng mô hình Hidden Markov kết hợp thuật toán Viterbi để gán nhãn cho từ loại Tiếng Việt

1.2. Phát biểu bài toán

Báo cáo trình bày nội dung chính liên quan đến lĩnh vực xử lý ngôn ngữ tự nhiên nói chung và xử lý ngôn ngữ Tiếng Việt nói riêng:

Bài toán Gán nhãn từ loại Tiếng Việt

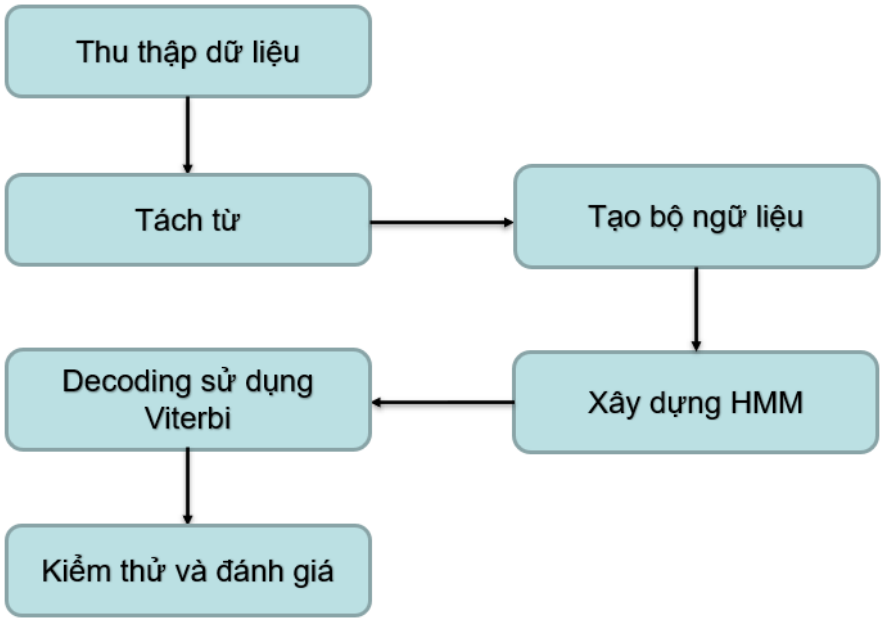
- **Input bài toán:** Đầu vào là một câu ngữ liệu bất kỳ bằng Tiếng Việt
- **Quá trình học:** Nhiệm vụ của chúng ta là Thực hiện xây dựng mô hình Hidden Markov thông qua các ma trận chuyển trạng thái (Transition Matrix) và ma trận thể hiện (Emission Matrix). Từ đó, sử dụng thuật toán Viterbi để gán nhãn dữ liệu
- **Output bài toán:** Cuối cùng, chúng ta đánh giá chất lượng của mô hình bằng việc sử dụng nó để dự đoán các nhãn cho một tập ảnh mà nó chưa thấy

trước đây. Sau đó chúng ta so sánh nhãn thật sự của những từ này với những nhãn được dự đoán bởi mô hình.

INPUT	OUTPUT
tìm tội phạm từ dấu vết hiện trường .	'V', 'N', 'E', 'N', 'N', 'CH'

Hình 1.1 Minh họa về INPUT và OUTPUT

- **Quy trình thực hiện**



Hình 1.2 Quy trình xử lý chung của đồ án

Trong phạm vi bài toán này, nhóm em tập trung thử nghiệm và tìm hiểu các phương pháp tách từ và gán nhãn từ đã học trên lớp kết hợp so sánh với các phương pháp tương ứng trong các thư viện có sẵn như: VnCoreNLP, Underseas,… Đây là tác vụ cơ sở cho các cho các bài toán nâng cao hơn như: Dịch máy (Translate Machine), Phân tích cú pháp cho câu,…

1.3. Thách thức bài toán

Bài toán gán nhãn từ loại Tiếng Việt cũng chứa rất nhiều các khó khăn mà chúng ta cần phải giải quyết và khắc phục.

Về dữ liệu, Với bộ ngữ liệu mà nhóm thu thập thì đều không thể đảm bảo về chất lượng cũng như số lượng. Điều này ảnh hưởng không nhỏ đến quá trinh huấn luyện và đánh giá kết quả của mô hình. Thêm vào đó, sự phân bố của các lớp nhãn từ loại có sự mất cân bằng lớn (imbalance classes).

Về xây dựng thủ công ngữ liệu, nhóm em phải cân nhắc rất kỹ các quy tắc để tách từ và gán nhãn bởi vì có rất nhiều từ ngữ viết tắt và trường hợp nhập nhằng gây khó khăn một phần trong quá trình thực hiện bài toán.

1.4. Cấu trúc báo cáo

Báo cáo này được trình bày trong chương, nội dung chính được tóm tắt như dưới đây:

- **Chương 1:** Giới thiệu chung về bài toán, động lực nghiên cứu, xác định bài toán và phạm vi tương ứng và thách thức của nó.
- **Chương 2:** Trình bày các cơ sở lý thuyết
- **Chương 3:** Trình bày kết quả thực nghiệm và đánh giá ưu điểm, hạn chế của các phương pháp được chọn để khảo sát. Xây dựng demo minh họa.
- **Chương 4:** Nêu kết quả đạt được, kết luận, định hướng nghiên cứu trong tương lai.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Tách từ

2.1.1. Lý do tách từ

Bài toán tách từ là một kỹ thuật không thể thiếu trong các bài toán xử lý ngôn ngữ tự nhiên. Mục đích của nó giúp chúng ta có thể phân tích một chuỗi từ thành các 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 em dùng ký tự '_' để biểu thị cho một từ có nhiều hơn một tiếng (từ ghép). Ví dụ:

"Tham gia giao thông phải an toàn ."

sẽ được tách như sau: "Tham_gia giao_thông phải an_toàn ."

2.1.2. Thuật toán Maximum Matching

Có rất nhiều thuật toán tách từ từ đơn giản đến phức tạp như: Maximum Matching, HMM, CRF,…

Ở đây, nhóm em sẽ thực hiện sử dụng 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 nhằn: 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”.

2.2. Mô hình Hidden Markov

2.2.1. Xích Markov (Markov Chain)

Xích Markov là một mô hình được sử dụng để mô tả một chuỗi các hành động và sự kiện diễn ra một cách liên tục, sao cho xác xuất của một sự kiện chỉ phụ thuộc vào sự kiện phía trước nó. Các sự kiện này chúng ta gọi là các trạng thái, còn giá trị xác xuất giữa các trạng thái gọi là các xác suất chuyển trạng thái (Transition Probability)

Hình 2.1 Minh họa Markov Chain

Ảnh minh họa ở trên: Các hình tròn màu xanh là các trạng thái, các mũi tên chứa xác xuất thể hiện cho xác xuất chuyển từ trạng thái này đến trạng thái kế tiếp.

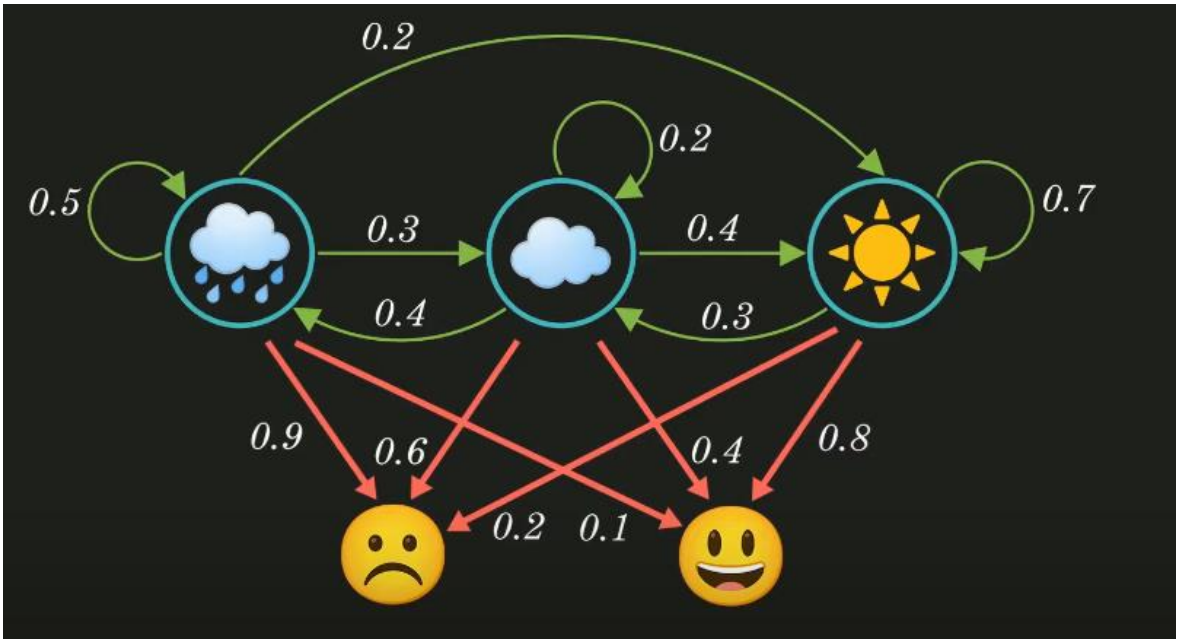
Đặc điểm của Markov Chain: là giả định các xác suất của các chuỗi là khi dự đoán trong tương lai chúng ta không quan tâm đến các trạng thái trong quá khứ mà chỉ phụ thuộc vào trạng thái hiện tại.Ví dụ: Ta muốn dự đoan thời tiết cho ngày mai, nếu dựa vào Markov Chain chúng ta chỉ quan tâm đến thời tiết của ngày hôm nay thay vì các ngày trước đó.

Tóm lại, Một Markov Chain sẽ bao gồm các thông tin sau đây:

- $X = x_1, x, x, \ldots x_N$: với X là tập hợp N trạng thái quan sát.
- Ma trân $A = a_{ii}$: Thể hiện cho xác xuất chuyển đổi giữa các trạng thái, ràng buộc $\sum_{j=1}^{n} a_{ij} = 1, \forall i.$

2.2.2. Mô hình Hidden Markov (HMM)

Mô hình Hidden Markov kế thừa nhiều ý tưởng của Markov Chain. Mô hình này được sử dụng rất phổ biến trong rất nhiều bài toán thuộc lĩnh vực Xử lý ngôn ngữ tự nhiên, ngòai gán nhãn từ loại có thể kể đến : bài toán tách từ, nhận diện giọng nói,…



Hình 2.2 Minh họa mô hình Hidden Markov

Mô hình Hidden Markov sử dụng một đồ thị có hướng thể hiện một số hữu hạn trạng thái ẩn và quan sát được tại một thời điểm. Một ma trận A (Transition Matrix) tương tự như Markov Chain thể hiện cho xác xuất chuyển đổi giữa trạng thái ẩn (Hidden

states) và một ma trận thể hiện B (Emission Matrix) mô tả cho xác xuất ta quan sát được khi ở một trạng thái nhất định

Khi áp dụng mô hình Hidden Markov này cho bài toán gán nhãn từ loại Tiếng Việt thì các nhãn từ loại chính là các trạng thái ẩn, các từ trong bộ ngữ liệu chính là trạng thái quan sát được của chúng ta

2.2.2.1. Transition Matrix A

Ma trận chuyển trạng thái A thể hiện xác xuất để chuyển một trạng thái sang trạng thái kế tiếp trong mô hình Markov Chain. Trong đó, mỗi hàng là xác suất chuyển từ trạng thái đại diện bởi hang đó sang các trạng thái khác. Vì vậy ta có ký hiệu như sau **A(x'|x)** thể hiện cho xác xuất chuyển từ trạng thái x sang x' trong ma trận A. Với A là ma trận chuyển đổi trạng thái, x là trạng thái hiện tại và x' là trạng thái có thể có trong tương lai.

2.2.2.2. Emission Matrix B

Ma trận thể hiện B mô tả xác xuất của một trạng thái quan sát được khi ta ở một trạng thái ẩn bất kỳ. Ta có thể ký hiệu như sau $B(W_i \mid x)$ thể hiện xác xuất của một trạng thái quan sát được Wi ứng với một trạng thai ẩn x trong ma trận B

2.3. Thuật toán Viterbi

Sau khi đã xây dựng được mô hình Hidden Markov, để có thể thực hiện tìm ra chuỗi trạng thái ẩn cho một chuỗi quan sát bất kỳ ta có thể sử dụng rất nhiều cách khác nhau:

+ Thực hiện Brute-Force để vét tất cả các trường hợp chuỗi trạng thái ẩn khác nhau, sau đo sẽ tinh xác xuất cho lần lượt các trường hợp này và tìm ra chuỗi xác xuất lớn nhất. Tuy nhiên, với sự chi phí tinh toán cao cùng với không chia sẻ các tài nguyên tinh toán, cách này trở nên bất khả thi với các trường hợp có quá nhiều trạng thái.

+ Ta sẽ khắc phục các nhược điểm của thuật toan vét cạn bằng sử dụng thuật toan Viterbi (Quy hoạch động) dựa trên 2 ma trận A và B. Quy trinh này được thực hiện với các bước sau:

- **Khởi tạo (Initialization)**: Khởi tạo 2 ma trận cùng chiều lưu trữ xác xuất (best_pro) và lưu trữ đường đi tốt nhất (best_paths)
- **Forward Viterbi**: Lần lượt cập nhật các giá trị của 2 ma trận này bằng cách tinh toán các xác xuất ở đường đi và đường đi tốt nhất tại từng thời điểm sử dụng 2 ma trận A,B
- **Backward Viterbi**: Thực hiện quay lui để tìm ra chuỗi trạng thái ẩn có xác xuất cao nhất

Chương 3. THỰC NGHIỆM

3.1. Thu thập dữ liệu

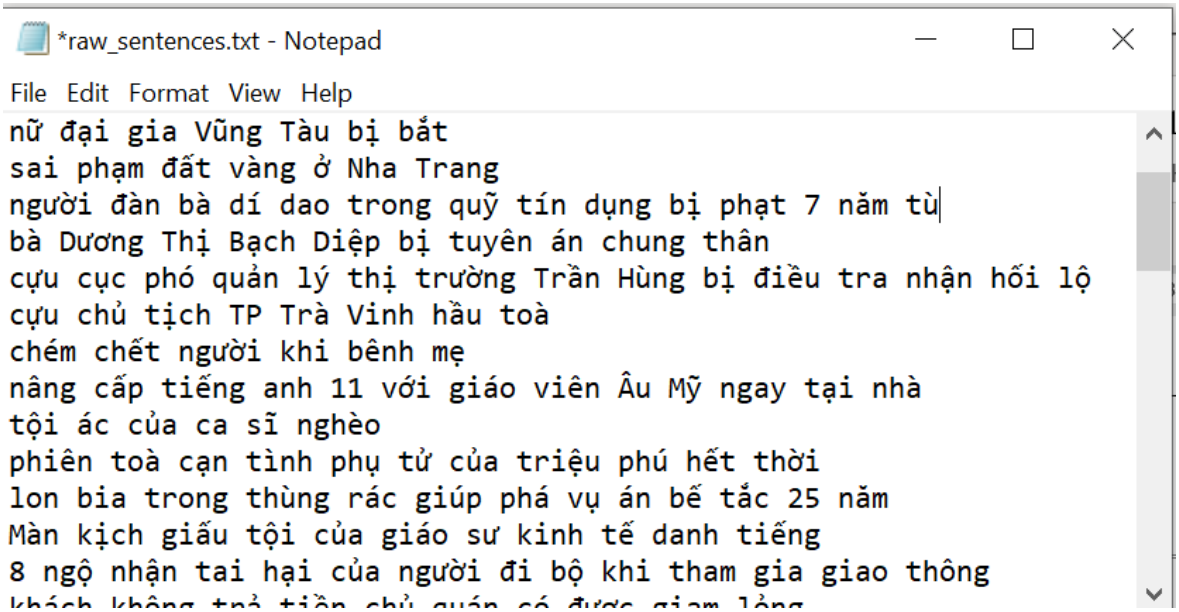
Nhóm tiến hành crawl dữ liệu từ các tiêu đề bài báo điện tử từ trang **https://vnexpress.net/.** Các bài báo tập trung chinh và chủ đề Đời sống và Pháp luật. Các thông tin chi tiết bao gồm:

- Số lượng câu: **56 câu**
- Mỗi dòng là 1 câu
- Các tiếng được phân tách bởi các khoảng trắng
- Tiến hành các bước tiền xử lý thủ công đơn giản: xóa dấu câu không cần thiết, viết thường các từ đầu câu,…
- Câu dài nhất: **15 tiếng**

  Bà Dương Thị Bạch Diệp nộp bí mật khoản vay 67000 lượng vàng cho tòa .
- Câu ngắn nhất: **3 tiếng**

  Cướp vé số .

Tất cả các câu được lưu vào trong files raw_sentences.txt



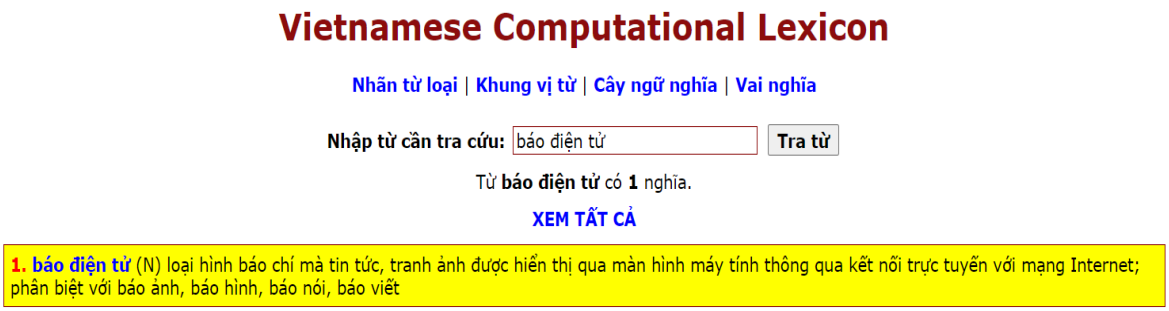
Hình 3.1 Dữ liệu thu thập

3.2. Tách từ

Nhóm em 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
- Sử dụng dấu **'.'** để ngăn cách các câu, kết thúc câu bằng ký tự xuống dòng **'\n'**
- Số lượng từ: **565 từ**
- Số lượng từ ghép: **156 từ**

Trong quá trình tách từ có xảy ra một số trường hợp nhập nhằng mà nhóm phải phân vân để lựa chọn cách tách từ thủ công. Các từ ghép này đều có nghĩa khi tra từ điển VLSP, tuy nhiên khi ta đem tách nó ra thì nó không hề bị mất đi ý nghĩa. Dưới đây là một số từ nhập nhằng:



Hình 3.2 Ví dụ trường hợp nhập nhằng

Từ **báo_điện_tử** là một trong số các trường hợp nhập nhằng, và sau khi bàn bạc nhóm em quyết định sẽ tách nó thành 2 từ riêng biệt là **báo** và **điện_tử**.

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=3**, kết hợp hai bộ từ điển bigram và trigram cùng với sự bổ sung các từ đã được gán nhãn thủ công từ trước **mydict.txt** để giúp cho thuật toán hoạt động tốt hơn. Kết quả được đánh giá cụ thể như sau:

	Accuracy
Maximum Matching	88.63%
VnCoreNLP	91.41%

Bảng 1. Kết quả tách từ

Dựa vào bảng kết quả ta thấy cả hai phương pháp tách từ đều tốt, trong đó thư viện VNCoreNLP cho kết quả tốt hơn. Nguyên nhân chính của việc này chính là bộ từ điển của nhóm thu thập vẫn còn nhiều thiếu sót, trong khi đó thư viện VNCoreNLP là một thư viện nổi tiếng trong lĩnh vực xử lý ngôn ngữ tự nhiên và đã được kiểm chứng bằng nhiều phương pháp. Nếu có dữ liệu mới xuất hiện, kết quả chắc chắn sẽ bị giảm.

3.3. Qui trình tạo ngữ liệu và gán nhãn

Sau khi đã thu được kết quả tách từ nhóm tiến hành thực hiện thảo luận và gán nhãn thủ công trên bộ dữ liệu. Trong quá trình gán nhãn, nhóm thực hiện vừa tra từ điển VLSP vừa chọn ra các trường hợp có nhiều nhãn nhập nhằng. 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. Các từ được gán nhãn tuân thủ theo quy tắc gán nhãn sau:

STT	Nhãn	Tên	Ví dụ
1	N	Danh từ	tiếng, nước, thủ đô, nhân dân, đồ đạc
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	mét, cân, giờ, nắm, nhúm, hào, xu
5	Ni	Danh từ ký hiệu	A1, A4, 60A, 60B, 20a, 20b, ABC
6	V	Động từ	ngủ, ngồi, cười; đọc, viết, đá, đặt, thích
7	A	Tính từ	tốt, xấu, đẹp; cao, thấp, rộng
8	P	Đại từ	tôi, chúng tôi, hắn, nó, y, đại nhân, đại
9	L	Định từ	mỗi, từng, mọi, cái; các, những, mấy
10	M	Số từ	một, mười, mười ba; dăm, vài, mươi
11	R	Phó từ	đã, sẽ, đang, vừa, mới, từng, xong, rồi
12	E	Giới từ	trên, dưới, trong, ngoài; của, trừ, ngoài
13	C	Liên từ	vì vậy, tuy nhiên, ngược lại
14	Cc	Liên từ đẳng lập	và, hoặc, với, cùng
15	I	Thán từ	ôi, chao, a ha
16	T	Trợ từ	à, a, á, ạ, ấy, chắc, chăng, cho, chứ
17	B	từ vay mượn	Internet, email, video, chat
18	Y	Từ viết tắt	OPEC, WTO, HIV
19	X	Các từ không thể phân loại	
20	Z	Yếu tố cấu tạo từ	bất, vô, phi
21	CH	Nhãn dành cho các loại dấu	. ! ? , ; :

Bảng 2. Quy ước gán nhãn

Có được bộ ngữ liệu gán nhãn, ta tiến hành đi chia **56 câu** thành hai phần là tập huấn luyện **(Train Set)** và tập kiểm thử **(Test Set)** với số lượng như sau: **40 câu** đầu cho tập Train và **16 câu** còn lại cho tập Test.

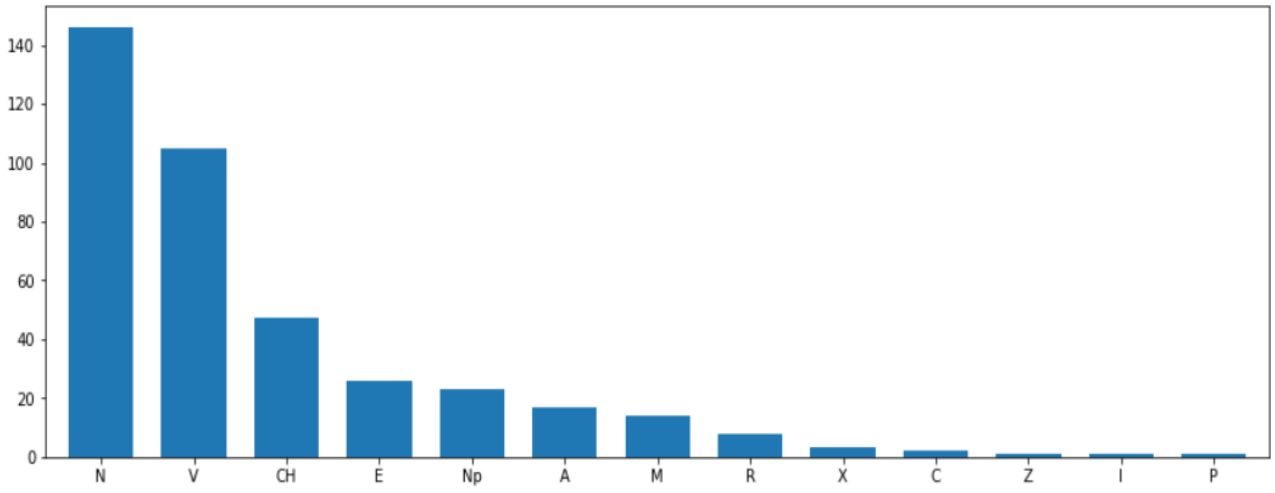
Ngoài ra, nhóm còn tiến hành xử lý ngữ liệu trước khi đưa vào mô hình Hidden Makov để huấn luyện. Cụ thể, mặc dù nhóm đã xây dựng bộ từ điển chứa ngữ liệu dựa trên dữ liệu của nhóm, tuy nhiên không tránh khỏi sẽ có trường hợp các từ trong tập test không nằm trong dữ liệu huấn luyện hay bộ từ điển mà ta đã xây dựng. Vì vậy nếu bất cứ từ nào không thuộc bộ từ vựng sẽ được thay đổi thành **"—unk—"** với nhãn được giữ nguyên như cũ. Điều này sẽ giúp thuật toán Viterbi có thể hoạt động tốt được với các trường hợp đặc biệt.

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:

- 1 file **manual_Tagging.txt** là ngữ liệu **56 câu** đã được gán nhãn thủ công, mỗi từ và nhãn ứng với một hàng, các từ và nhãn được ngăn cách bởi ký tự **"\t"**
- 2 file **train.txt** và **train_notag.txt** có cấu trúc tương tự với file gán nhãn thủ công, trong đó file train.txt chứa các từ và nhãn để tính toán các thông tin để huấn luyện cho mô hình Hidden Markov, còn file **train_notag.txt** là ngữ liệu huấn luyện nhưng không chứa các nhãn chủ yếu để kiểm tra được độ chính xác của mô hình xây dựng
- 2 file **test.txt** và **test_notag.txt** cũng tương tự như vậy, ý nghĩa của tập test này để kiểm tra tính chính xác của mô hình khi dự đoán trên tập dữ liệu không được mô hình nhìn thấy, trong đó **test.txt** được xem là nhãn thực đã được gán thủ công (ground-truth) còn **test_notag.txt** là được sử dụng thực hiện dự đoán

Tập train chứa các thông tin như sau: **số câu: 40, số nhãn: 408.** Cụ thể:

	N	V	CH	E	Np	A	M	R	X	C	Z	I	P	Total
0	159	105	47	26	23	17	14	9	3	2	1	1	1	408



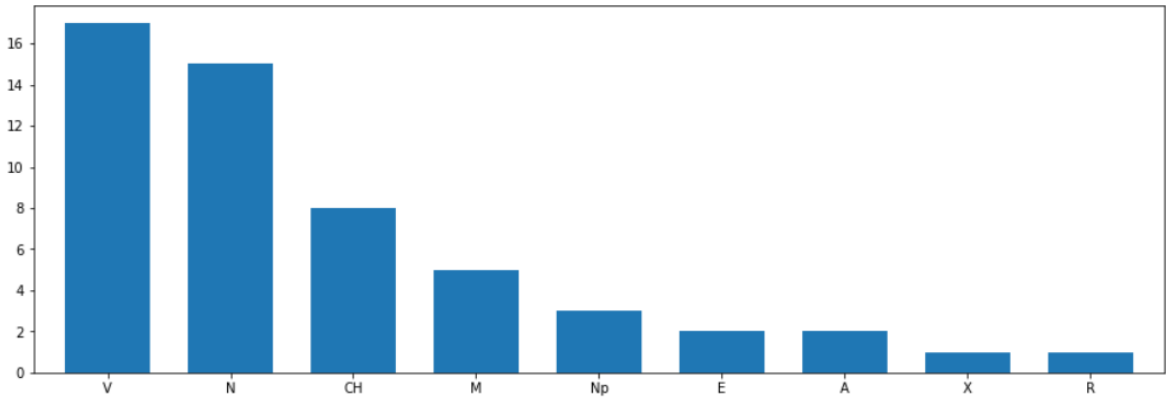
Hình 3.3 Phân bố dữ liệu trên tập Train

Dựa vào trực quan hóa bộ dữ liệu huấn luyện, nhóm nhận định rằng mô hình xây dựng sẽ khó có thể hoạt động tốt trên thực tế, kết quả sẽ không cao. Nguyên nhân chủ yếu là:

- Bộ dữ liệu phân bố giữa các nhãn không đồng đều, và bị mất cân bằng trầm trọng
- Bộ dữ liệu đem đi huấn luyện chi có 40 câu là quá ít, không thể bao quát hết cho tất cả các trường hợp

Tập train chứa các thông tin như sau: **số câu: 10, số nhãn: 57.** Cụ thể:

	N	V	CH	M	A	Np	E	X	R	Total
0	17	17	8	5	3	3	2	1	1	57



Hình 3.4 Phân bố dữ liệu trên tập Test

3.4. Xây dựng mô hình Hidden Markov

3.4.1. Tính toán các thông tin cần thiết

Trước khi đi vào xây dựng 2 ma trận A, B của mô hình Hidden Markov, nhóm em tiến hành tính Transition Count và Emission Count và lưu lại sử dụng cấu trúc dữ liệu từ điển (dictionary). Bởi vì để mô hình hoạt động tốt thì cần phải có một nhãn gọi là nhãn bắt đầu câu, vì vậy trong nhóm sẽ thêm vào danh sách các nhãn một nhãn kí hiệu là **'—s—'** để đánh dấu vị trí bắt đầu câu.

Từ điển Transition Count có dạng như sau:

```
(('--s--', 'A'), 1)
(('A', 'N'), 3)
(('N', 'Np'), 12)
```

Trong đó, mỗi dòng của từ điển sẽ đếm số lần chuyển trạng thái của các nhãn trong bộ dữ liệu. Cụ thể, ở dòng đầu số 1 trên hình cho ta biết được số lần chuyển trạng thái từ nhãn bắt đầu **'—s—'** sang nhãn **'A'** là 1 lần

Từ điển Emission Count có dạng như sau:

```
(('A', 'nhiều'), 1)
(('N', 'người'), 4)
(('Np', 'Hà_Nội'), 1)
```

Mỗi dòng của từ điển Emission Count đếm số lần mà một từ trong ngữ liệu được gán cho một nhãn trong bộ dữ liệu. Ví dụ ở dòng thứ 2 của từ điển, ta biết được trong bộ ngữ liệu của chúng ta sẽ có 4 lần mà từ **'người'** được gán nhãn là **'N'**

Ngoài ra, nhóm còn tính số lần xuất hiện của các nhãn trong tập dữ liệu với số liệu cụ thể như sau:

{**'A'**: 17, **'N'**: 146, **'Np'**: 23,

'V': 105, **'CH'**: 47, **'--s--'**: 46,

'E': 26, **'M':** 14, **'Z'**: 1, **'X'**: 3,

'I': 1, **'R'**: 8, **'C'**: 2, **'P'**: 1}

3.4.2. Xây dựng ma trận chuyển đổi trạng thái A (Transition Matrix)

Dựa vào các thông tin đã tính toán ở trên nhóm tiến hành xây dựng ma trận chuyển đổi trạng thái A với công thức sau:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i) + \alpha}{C(t_{i-1}) + \alpha * N}$$

Hình 3.5 Công thức cho ma trận A

Trong đó:

- **$P(t_i|t_{i-1})$** là xác xuất chuyển trạng thái từ nhãn i-1 sang nhãn i
- **$C(t_{i-1})$** là số lượng nhãn i-1 đã được tính từ trước

- **$C(t_{i-1}, t_i)$** là số lần xuất hiện nhãn i-1 kéo theo nhãn i trong bộ ngữ liệu, đã được tính sẵn trong từ điển Transition Count
- **α** là tham số smoothing được sử dụng để làm cho các giá trị trong ma trận A không chứ các giá trị không xác định. Phương pháp sử dụng là smoothing Laplace
- **N** là số lượng nhãn

	A	C	CH	E	I	M	N	Np	P	R	V	X	Z
--s--	0.021754	2.17E-05	2.17E-05	2.17E-05	2.17E-05	0.065219	0.543335	0.086952	2.17E-05	0.021754	0.239079	0.021754	0.021754
A	5.88E-05	0.117609	0.235159	0.117609	5.88E-05	0.117609	0.176384	5.88E-05	5.88E-05	0.058834	0.176384	5.88E-05	5.88E-05
C	0.000497	0.000497	0.000497	0.000497	0.000497	0.000497	0.000497	0.000497	0.000497	0.000497	0.993545	0.000497	0.000497
CH	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05	2.13E-05
E	3.84E-05	3.84E-05	3.84E-05	3.84E-05	3.84E-05	3.84E-05	0.730414	0.153802	3.84E-05	0.038479	0.07692	3.84E-05	3.84E-05
I	0.000986	0.000986	0.000986	0.987179	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986
M	7.14E-05	7.14E-05	7.14E-05	0.071429	7.14E-05	0.142786	0.713643	7.14E-05	7.14E-05	7.14E-05	0.071429	7.14E-05	7.14E-05
N	0.054796	6.85E-06	0.171223	0.123283	6.85E-06	0.006856	0.301348	0.082191	6.85E-06	0.027401	0.226013	0.006856	6.85E-06
Np	4.35E-05	4.35E-05	0.347658	4.35E-05	0.043495	4.35E-05	4.35E-05	4.35E-05	0.043495	4.35E-05	0.564917	4.35E-05	4.35E-05
P	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986	0.987179	0.000986	0.000986	0.000986
R	0.124906	0.000125	0.000125	0.000125	0.000125	0.000125	0.000125	0.000125	0.000125	0.000125	0.873596	0.000125	0.000125
V	0.066667	9.52E-06	0.085712	0.0381	9.52E-06	0.057145	0.409479	0.019055	9.52E-06	9.52E-06	0.314253	0.009532	9.52E-06
X	0.000332	0.000332	0.332117	0.000332	0.000332	0.000332	0.332117	0.332117	0.000332	0.000332	0.000332	0.000332	0.000332
Z	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986	0.987179	0.000986	0.000986	0.000986	0.000986	0.000986	0.000986

Hình 3.6 Minh họa của ma trận A

Ma trận A có các hàng và các cột là các nhãn trong bộ ngữ liệu, mỗi ô a_{ij} thể hiện cho xác xuất chuyển trạng thái từ nhãn ở hàng i sang nhãn của cột j. Trong đó mỗi hàng phải có tổng các xác suất cộng lại bằng 1

3.4.3. Xây dựng ma trận thể hiện B (Emission Matrix)

Dựa vào các thông tin đã tính toán ở trên nhóm tiến hành xây dựng ma trận chuyển đổi trạng thái A với công thức sau:

$$P(w_i|t_i) = \frac{C(t_i, word_i) + \alpha}{C(t_i) + \alpha * N}$$

Hình 3.7 Công thức cho ma trận A

Trong đó:

- **$P(w_i|t_i)$** là xác xuất để một từ thứ i được gán nhãn là t_i
- **$C(t_i)$** là số lượng nhãn i đã được tính từ trước

- **C(t_i, w_i)** là số lần xuất hiện của một từ thứ i trong bộ ngữ liệu được gán là nhãn t_i, đã được tính từ trước trong từ điển Emission Count
- **α** là tham số smoothing được sử dụng để làm cho các giá trị trong ma trận **B** không chứ các giá trị không xác định. Phương pháp sử dụng là smoothing Laplace
- **N** là số lượng từ trong từ điển

	tìm	tội_phạm	hiện_trường	từ	dấu_vết
A	0.000014	0.000014	0.000014	0.000014	0.000014
C	0.000018	0.000018	0.000018	0.000018	0.000018
CH	0.000010	0.000010	0.000010	0.000010	0.000010
E	0.000012	0.000012	0.000012	0.012383	0.000012
I	0.000018	0.000018	0.000018	0.000018	0.000018
M	0.000015	0.000015	0.000015	0.000015	0.000015
N	0.000005	0.004935	0.004935	0.000005	0.004935
Np	0.000013	0.000013	0.000013	0.000013	0.000013
P	0.000018	0.000018	0.000018	0.000018	0.000018
R	0.000016	0.000016	0.000016	0.000016	0.000016
V	0.006263	0.000006	0.000006	0.000006	0.000006
X	0.000017	0.000017	0.000017	0.000017	0.000017
Z	0.000018	0.000018	0.000018	0.000018	0.000018

Hình 3.8 Minh họa một phần ma trận B

Ở trên là một phần ma trận thể hiện B được cắt ra. Bởi vì số lượng từ trong ngữ liệu lớn nên để đưa toàn bộ ma trận B vào sẽ gây mất thẩm mỹ. Dựa vào ma trận B trên ta có thể thấy rằng ở các từ được gán nhãn sẽ có xác xuất tại nhãn đó lớn hơn so với các nhãn còn lại

3.5. Thuật toán Viterbi

Như vậy, sau khi tính xong 2 ma trận A, B thì cơ bản chúng ta đã hoàn thành việc xây dựng mô hình Hidden Markov ứng với bộ ngữ liệu của chúng ta. Việc tiếp theo chúng ta sẽ kết hợp mô hình Hidden Markov với thuật toán Viterbi để dự đoán và thử gán nhãn cho dữ liệu test. Nguyên nhân mà chúng ta sử dụng nó cũng như về quy

trình tổng quát đã được nhóm trình bày tại phần Chương 2. Cơ sở lý thuyết. Bây giờ nhóm sẽ đi vào cụ thể từng bước áp dụng cho bộ ngữ liệu.

Đầu tiên, ta sẽ khởi tạo 2 ma trận cùng chiều là best_prob và best_path lưu giữ các thông tin về xác xuất mà một từ ứng với một nhãn bất kỳ, và thông tin đường đi tốt nhất

Tiếp đến, sử dụng 2 ma trận A, B (tức mô hình Hidden Markov) đã được xây dựng từ trước để tiến hành cập nhật các ô trong ma trận. Cụ thể:

- Lặp qua tất cả các từ trong ngữ liệu, với mỗi từ ta thực hiện điền các ô của ma trận best_pro và best_path bằng công thức:

$$\mathbf{best\_pro_{wordi,\ tag} = best\_prob_{wordi-1,\ k} + log(A_{k,\ tag}) + log(B_{tag,\ wordi})}$$

Hình 3.9 Công thức tính xác xuất thành phần

Trong đó:

+ $\mathbf{best\_pro_{wordi,\ tag}}$ là xác suất của từ trong ngữ liệu ứng với một nhãn bất kỳ

+ $\mathbf{best\_prob_{wordi-1,\ k}}$ xác suất của từ phía trước của từ hiện tại với nhãn là k

+ $\mathbf{A_{k,\ tag}}$ là xác xuất chuyển từ nhãn phía trước **k** sang nhãn hiện tại là tag trong ma trận A

+ $\mathbf{B_{tag,\ wordi}}$ là xác xuất để từ đang xét $word_i$ được gán là nhãn tag trong ma trận **B**

+ Nhóm em sử dụng log ở đây là để làm việc với các con số xác xuất nhỏ, tránh hiện tượng tràn số khi thực hiện nhân lần lượt các xác xuất nhỏ với nhau với các câu quá dài

- Trong quá trình tính toán, ta sẽ lưu lại được từ hiện tại với nhãn có giá trị xác xuất lớn nhất và lưu vào vào ma trận **best_prob**

- Đồng thời lúc này ta sẽ lưu lại vị trí k (tức là nhãn trước đó) và cập nhật ma trận **best_path** tại cùng vị trí ứng với ma trận **best_prob**

  Cuối cùng, sau khi đã hoàn thành xong 2 ma trận **best_pro** và **best_path**, ta sẽ tiến hành sử dụng 2 ma trận này để quay lui trả về giá trị dự đoán nhãn của câu. Cụ thể, ta tiến hành truy vết từ vị trí cột cuối cùng có giá trị xác xuất lớn nhất của ma trận đường đi **best_path**. Tại vị trí này, nó sẽ lưu vị trí trỏ tới vị trí ở các cột phía trước của nó miễn là giá trị con trỏ tới không bằng 0.

  Dưới đây, nhóm tiến thực hiện minh họa thuật toán Viterbi với một câu chuỗi cho sẵn như sau:

  "tham gia giao thông phải an toàn ."

  Kết quả của hai ma trận xác xuất và đường đi tốt nhất được cập nhật như sau:

	tham_gia	giao_thông	phải	an_toàn	.
--s--	-15.007303487850837	-26.181006896355157	-35.65476696107309	-46.139699493499414	-58.163394980689894
A	-14.313627770868809	-25.036971326740122	-36.02633950925827	-47.212306587077705	-59.30028705826397
C	-15.006501460867032	-27.11613937785824	-38.22329059603277	-49.273871323720236	-60.45983840153967
CH	-15.007321708263166	-24.481081772981106	-34.96601430540743	-46.98970979259791	-54.36876084350523
E	-15.006939009870386	-24.794356633443364	-35.279289165869685	-47.46257981686027	-58.55323671487538
I	-15.006483225503189	-27.116121142494393	-38.22327236066893	-49.291456402831976	-61.1461827190026
M	-13.62042589817996	-25.170448027235743	-37.3372443580669	-47.345783287573326	-59.51257961840448
N	-11.751027349727694	-22.235959882154013	-34.4192505331446	-45.509907431159704	-57.67670376199086
Np	-13.397446414279594	-25.17379157199158	-35.6587241044179	-47.842014755408485	-58.93267165342358
P	-15.006483225503189	-27.116121142494393	-38.22327236066893	-49.291456402831976	-61.1461827190026
R	-14.31346368550759	-26.129029556372856	-36.61396208879917	-48.79725273978976	-59.88790963780487
V	-11.424858986101777	-23.591655316932933	-33.60019424643936	-45.76699057727051	-57.933786908101666
X	-14.313372515338406	-26.42301043232961	-37.53016165050414	-48.598345692667195	-60.45307200883782
Z	-14.313336044943245	-27.116121142494393	-38.22327236066893	-49.291456402831976	-60.45303553844266

Hình 3.10 Minh họa ma trận best_pro

	tham_gia	giao_thông	phải	an_toàn	.
--s--	0	3	3	3	3
A	0	11	7	11	7
C	0	11	7	1	1
CH	0	7	7	11	7
E	0	7	7	7	7
I	0	11	7	11	0
M	0	11	11	11	11
N	0	11	7	11	11
Np	0	7	7	7	7
P	0	11	7	11	0
R	0	7	7	7	7
V	0	11	7	11	11
X	0	11	7	11	0
Z	0	11	7	11	0

Hình 3.11 Minh họa ma trận best_path

Dựa vào 2 ma trận ở trên ta xác định được chuỗi nhãn có khả năng lớn nhất là:

“tham_gia/V giao_thông/N phải/V an_toàn/N ./CH”

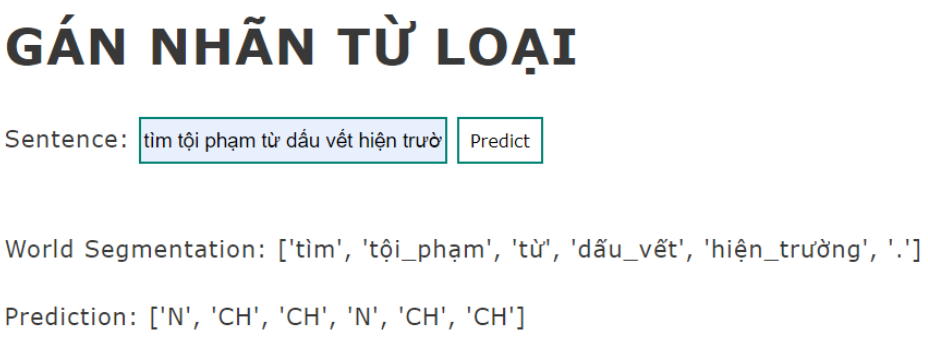
3.6. Demo minh họa

3.5.1. Framework Flask Python

Flask là một web framework, được xây dựng dựa trên ngôn ngữ lập trình python. Sử dụng flask, chúng ta có thể xây dựng một ứng dụng web từ đơn giản đến nâng cao. Điểm đặc biệt của Flask là nó sử dụng một cách độc lập mà không cần phải nhờ đến sự hỗ trợ đến từ các thư viện bên ngoài khác. Ngoài ra, Flask còn có ưu điểm là nhẹ và dễ sử dụng và tinh chỉnh. Nhóm quyết định sẽ xây dựng một ứng dụng phân loại dựa trên framework Flask.

3.5.2. Giao diện và Chức năng chính của Web App

Hình 3.12 Minh họa giao diện của app



Hình 3.13 Minh họa kết quả trên app

- **Sentence:** Câu nhập vào
- **Word Segmentation:** Từ được tách trong câu nhập vào
- **Prediction:** Nhãn dự đoán của các từ trong câu

Chương 4. ĐÁNH GIÁ, KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Đánh giá

Ta tiến hành đánh đánh giá hiệu suất của mô hình trên các tập huấn luyện và tập kiểm thử. Trong đó, ngoài sử dụng mô hình Hidden Markov đã được xây dựng từ trước, nhóm cũng sử dụng thư viện có sẵn VnCoreNLP để gán nhãn và so sánh.

Phương pháp đánh giá được thực hiện như sau:

- Xây dựng bộ gán nhãn từ loại dựa trên các mô hình và thư viện có sẵn trên các tập dữ liệu mất nhãn (train_notag.txt và test_notag.txt) để dự đoán chuỗi nhãn
- So sánh kết quả tách từ của chuỗi nhãn dự đoán và chuỗi nhãn thực tế đã gán thủ công (tập Gold)

4.1.1. Độ đo

Độ đo để đánh giá nhóm sử dụng là Accuracy (độ chính xác) với Acc = n/N. Trong đó:

+ N là tổng số lượng nhãn ứng với các từ có trong ngữ liệu gán nhãn thủ công

+ n là số lượng nhãn được dự đoán đúng từ mô hình

4.1.2. Kết quả

Dựa trên phương pháp và độ đo đánh giá đã định nghĩa, nhóm tiến hành cài đặt và cho ra bảng kết quả sau:

	Accuracy
Hidden Markov (Train)	80.46%
Hidden Markov (Test)	48.15%
VnCoreNLP (Train)	92.89%
VnCoreNLP (Test)	94.44%

Bảng 3. Bảng kết quả thực nghiệm

Kết quả dự đoán trên tập test sử dụng 2 phương pháp được lưu trong thư mục data_tag với các tên là **VnCoreNLP_test_predict.txt** và **HMM_Viterbi_test_predict.txt.**

Một câu ví dụ:

Thủ công:

màn_kịch/N giấu/V tội/N của/E giáo_sư/N kinh_tế/N danh_tiếng/A ./CH

HMM + Viterbi:

màn_kịch/N **giấu/CH** tội/N của/E giáo_sư/N kinh_tế/N **danh_tiếng/N** ./CH

VnCoreNLP:

màn_kịch/N giấu/V tội/N của/E giáo_sư/N kinh_tế/N **danh_tiếng/N** ./CH

Nhận xét:

Kết quả accuracy của mô hình Hidden Markov của chúng em thấp hơn VnCoreNLP ở cả tập train (80.46% < 92.89%) và tập test (48.15% < 94.44%). Với kết quả accuracy ở tập test (48.15%) thấp hơn khá nhiều so với ở tập train, chứng tỏ rằng mô hình Hidden Markov của chúng em đang bị overfitting.

Nguyên nhân chính dẫn đến kết quả trên là do ngữ liệu chung em thu thập còn ít chỉ sử dụng có 40 câu để huấn luyện, chưa bao quát hết được ngữ cảnh (ngữ liệu thu thập tập trung vào chủ đề đời sống – pháp luật) , phân bố nhãn từ loại không đều, ngoài ra bộ từ điển của nhóm xây dựng cũng không đầy đủ, khiến việc nhược điểm trong quá tách từ vẫn chưa khắc phục được.

4.2. Kết luận và hướng phát triển

Trong đồ án này, nhóm em đã thực hiện tìm hiểu và giải quyết các vấn đề cơ bản của bài toán gán nhãn từ loại Tiếng Việt sử dụng mô hình thống kê là Hidden Markov. Trong đó, hiểu được quy trình thực hiện của một bài toán trong lĩnh vực Xử lý ngôn ngữ tự nhiên, từ thu thập ngữ liệu, xử lý tách từ sử dụng Maximum Matching đến xây dựng mô hình Hidden Markov và sử dụng thuật toán Viterbi để gán nhãn cho từ loại

Tiếng Việt. Ngoài ra, qua đồ án này nhóm cũng có thể hiểu được tầm quan trọng của dữ liệu và các phương pháp đánh giá cho các bài toán Xử lý ngôn ngữ tự nhiên nói riêng và các ngành liên quan đến dữ liệu nói chung.

Cuối cùng, nhóm em cũng đề xuất các ý tưởng cải thiện để phát triển sâu hơn cho đồ án này. Cụ thể, nhóm có thể sẽ thu thập thêm nhiều các dữ liệu từ các nguồn khác nhau, đa dạng các lĩnh vực, chủ đề và từ loại hoặc xây dựng các phướng pháp mới thay thế cho thuật toán tách từ Maximum Matching quá đơn giản và bị phụ thuộc vào nhiều yếu tố như: sử dụng Mô hình Hidden Markov, …. Từ các hướng phát triển này nhóm mong muốn có thể sử dụng đồ án này để phát triển tiếp cho các bài toán thực tế.

LỜI CẢM ƠN

Kết thúc môn học, nhóm em xin gửi lời cảm ơn đến cho thầy Nguyễn Trọng Chỉnh. Mặc dù trong quá trình làm việc chúng em có kết quả không được như mong muốn, nhưng chúng em vẫn rất biết ơn thầy đã giảng dạy và hướng dẫn chúng em trong quá trình học tập và đưa ra các nhận xét khách quan để nhóm hoàn thiện đồ án. Tụi em chúc thầy và gia đình luôn luôn mạnh khỏe và hạnh phúc trong mùa dịch này, mong sẽ có thể tiếp tục làm việc với thầy trong các môn học khác

TÀI LIỆU THAM KHẢO

https://github.com/UITTrinhQuangTruong/CS221.L11

https://github.com/18520339/vietnamese-pos-tagging

https://github.com/alejandropuerto/viterbi-HMM-POS-tagging