ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC

CS221 – XỬ LÝ NGÔN NGỮ TỰ NHIÊN

GÁN NHÃN TỪ LOẠI BẰNG MÔ HÌNH HIDDEN MARKOV

STT	MSSV	Họ và tên	Email liên hệ
1	20521360	Châu Phạm Quốc Hưng	20521360@gm.uit.edu.vn
2	20521363	Lê Quang Hùng	20521363@gm.uit.edu.vn

Giảng viên hướng dẫn: ThS. Nguyễn Trọng Chỉnh

Nhóm sinh viên thực hiện: CS221.M12.KHCL

THÔNG TIN CHUNG

Đề tài:	Gán nhãn từ loại bằng mô hình Hidden Markov.
Môn học:	CS221 – Xử lý ngôn ngữ tự nhiên.
Lớp:	CS221.M12.KHCL.
Giảng viên hướng dẫn:	ThS. Nguyễn Trọng Chỉnh.
Thời gian thực hiện:	Học kỳ 1. Năm học: 2021 – 2022.
Sinh viên thực hiện:	Châu Phạm Quốc Hưng – 205211360. Lê Quang Hùng – 20521363.
Nội dung đề tài:	Gán nhãn từ loại tiếng Việt bằng mô hình Hidden Markov. Hay nói cách khác nhóm sẽ xác định các chức năng ngữ pháp của từ trong câu sử dụng Hidden Markov Model, 1 mô hình thống kê thường được dùng trong việc quan sát các chuỗi sự kiện có liên quan nhau. Đưa ra kết quả thử nghiệm từ bộ ngữ liệu mà nhóm

	đã thu thập. So sánh độ chính xác qua các hướng tiếp cận khác nhau.
Kế hoạch thực hiện:	Tuần 1, 2: Thành lập nhóm. Chọn đề tài. Xây dựng kế hoạch thực hiện và phân công nhiệm vụ. Tuần 3, 4, 5, 6, 7: Khảo sát tình hình thực tế. Thu thập các thông tin liên quan và tìm nguồn tài liệu tham khảo. Tìm hiểu các kiến thức cần thiết cho quá trình thực hiện đề tài. Tuần 8, 9, 10, 11, 12: Tìm hiểu các công nghệ và công cụ có thể sử dụng trong đề tài. Thiết kế, xây dựng và viết chương trình cài đặt. Đánh giá và kiểm thử chương trình cài đặt. Tuần 13, 14: Chuẩn bị slide trình bày và tập duyệt báo cáo. Tuần 15: Báo cáo đồ án. Tuần Dự trữ: Chỉnh sửa báo cáo và chương trình cài đặt sau khi báo cáo.

LỜI CẢM ƠN

Nhóm xin chân thành gửi lời cảm ơn đến ThS. Nguyễn Trọng Chỉnh – Giảng viên khoa Khoa học máy tính, Trường Đại học Công nghệ thông tin, Đại học Quốc gia thành phố Hồ Chí Minh, đồng thời là giản viên giảng dạy lớp CS221.M12.KHCL – Môn Xử lý ngôn ngữ tự nhiên, trong thời gian đã tận tình hướng dẫn và định hướng cho nhóm trong suốt quá trình thực hiện và hoàn thành đồ án.

Trong quá trình thực hiện đồ án nhóm đã cố gắng rất nhiều để hoàn thành đồ án một cách tốt nhất và hoàn thiện nhất, song cũng sẽ không tránh khỏi được những sai sót ngoài ý muốn. Nhóm mong rằng sẽ nhận được những lời nhận xét và những lời góp ý chân thành từ quý thầy/cô và các bạn trong quá trình thực hiện chương trình của nhóm để chương trình ngày càng hoàn thiện hơn. Mọi thắc mắc cũng như mọi góp ý của mọi người xin gửi email về một trong các địa chỉ email sau: 20521360@gm.uit.edu.vn (Châu Phạm Quốc Hưng), 20521363@gm.uit.edu.vn (Lê Quang Hùng). Mỗi ý kiến đóng góp sẽ là một nguồn động lực to lớn đối với nhóm để nhóm có thể cố gắng cải tiến chương trình ngày càng hoàn thiện và phát triển đồ án lên một mức cao hơn, nhóm cũng sẽ dựa vào đó để phát triển hơn những ưu điểm và cải thiện được phần nào đó những nhược điểm của chương trình. Hy vọng đề tài "Gán nhãn từ loại bằng mô hình Hidden Markov Model" do nhóm thực hiện sẽ trở thành một công cụ hữu ích và có thể ứng dụng được trong lĩnh vực Xử lý ngôn ngữ tự nhiên.

Thành phố Hồ Chí Minh, tháng 1 năm 2022

Nhóm sinh viên thực hiện

Châu Phạm Quốc Hưng, Lê Quang Hùng

MỤC LỤC

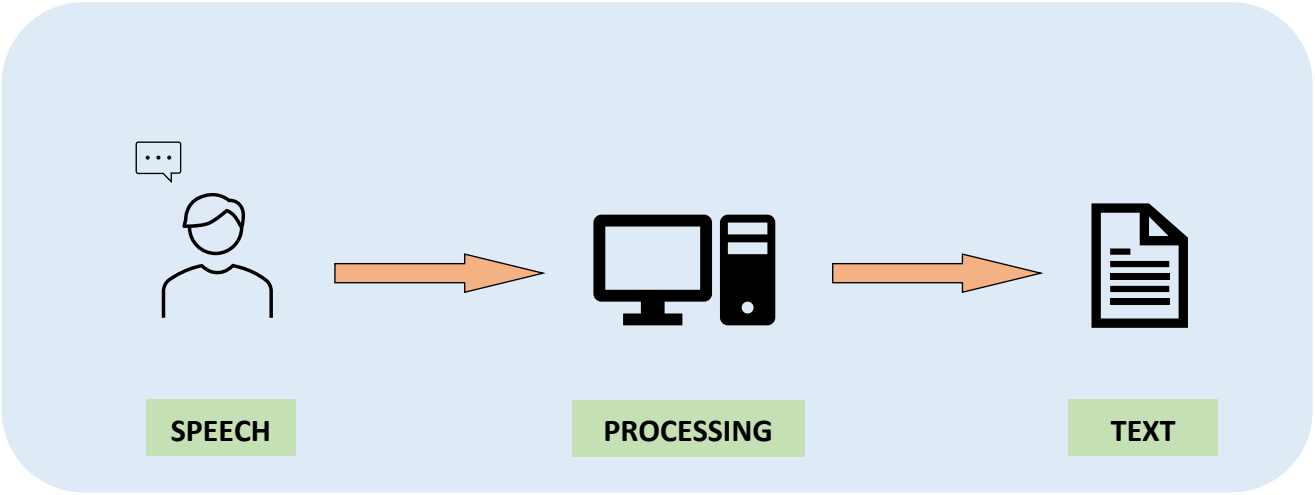
Trang

Chương 1. TỔNG QUAN

1. Giới thiệu

Khái niệm xử lý ngôn ngữ tự nhiên

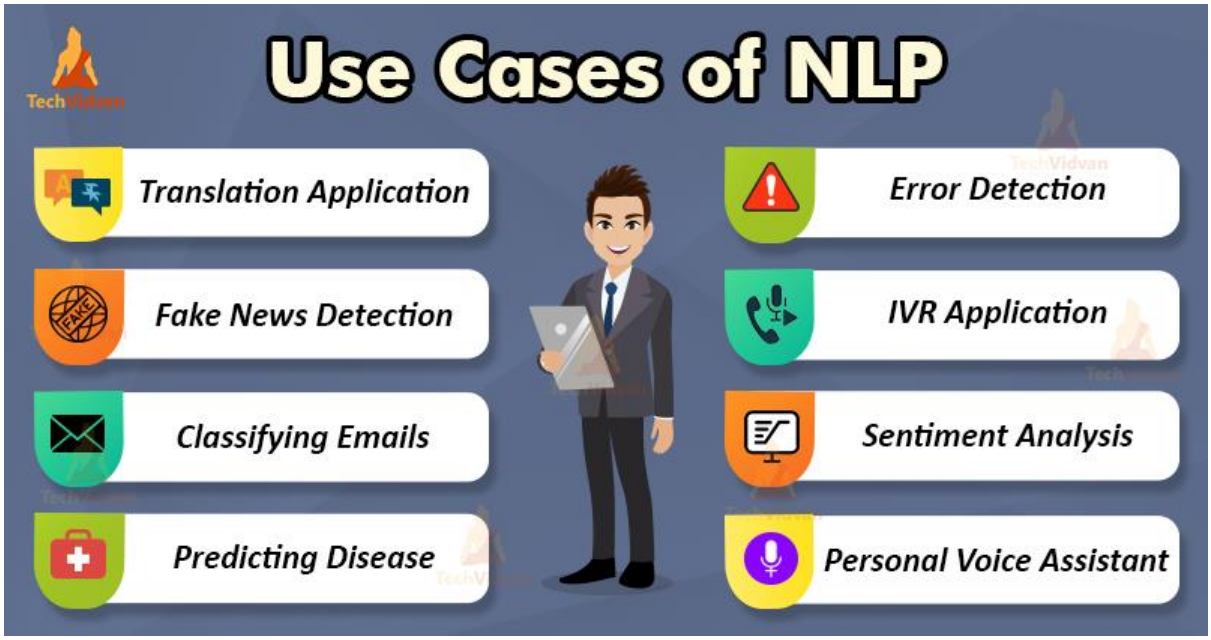
Xử lý ngôn ngữ tự nhiên (Natural language processing) là một nhánh trong ngành khoa học máy tính, nghiên cứu và áp dụng máy tính vào việc nhận dạng và phân tích được ngôn ngữ tự nhiên như văn bản và tiếng nói thông dụng. Mục đích của những người nghiên cứu về lĩnh vực này là tận dụng những kiến thức về ngôn ngữ học để phát triển những công cụ và kỹ thuật phù hợp vào hệ thống máy tính sao cho máy tính có thể giải quyết những vấn đề thực tế được đặt ra có liên quan đến ngôn ngữ tự nhiên. Tất cả có thể được thực hiện trên quy mô như câu trong văn bản, lời nói hay cả trong những bài văn, bài thuyết trình và hơn thế nữa.



Mô phỏng quá trình áp dụng xử lý ngôn ngữ tự nhiên trong việc chuyển đổi lới nói thành văn bản (Text-to-Speech)

Nói ngắn gọn, xử lý ngôn ngữ tự nhiên là làm sao cho máy tính có thể hiểu và xử lý được ngôn ngữ của con người, ví dụ như tiếng Anh, Việt, Nga, Trung, … Đầu vào có thể là văn bản chữ viết, tiếng nói.

Xử lý ngôn ngữ tự nhiên bao gồm cả khoa học dữ liệu, khoa học máy tính và ngôn ngữ học, ứng dụng được cả trong lý thuyết lẫn thực tế. Hiện nay đang có nhiều nền tảng dễ dàng truy cập như MonkeyLearn, cung cấp các công cụ hỗ trợ áp dụng xử lý ngôn ngữ tự nhiên có thể giúp đỡ doanh nghiệp xử lý lượng lớn dữ liệu văn bản, đẩy nhanh chiến dịch, giảm chi phí, tăng sự hài lòng của khách hàng và hơn thế nữa.



Một số ứng dụng của xử lý ngôn ngữ tự nhiên hiện nay

Các cơ sở khoa học của xử lý ngôn ngữ tự nhiên

Ngôn ngữ: Là một hệ thống các dấu hiệu có chức năng như là một phương tiện của sự tiếp xúc, một công cụ của tư duy. Nó là một hiện tượng tồn tại khách quan trong đời sống tinh thần của xã hội, là một hiện tượng của nền văn hóa tinh thần phản ánh trong ý thức cộng đồng và trừu tượng hóa khỏi bất kỳ một tư tưởng, cảm xúc và ước muốn cụ thể nào.

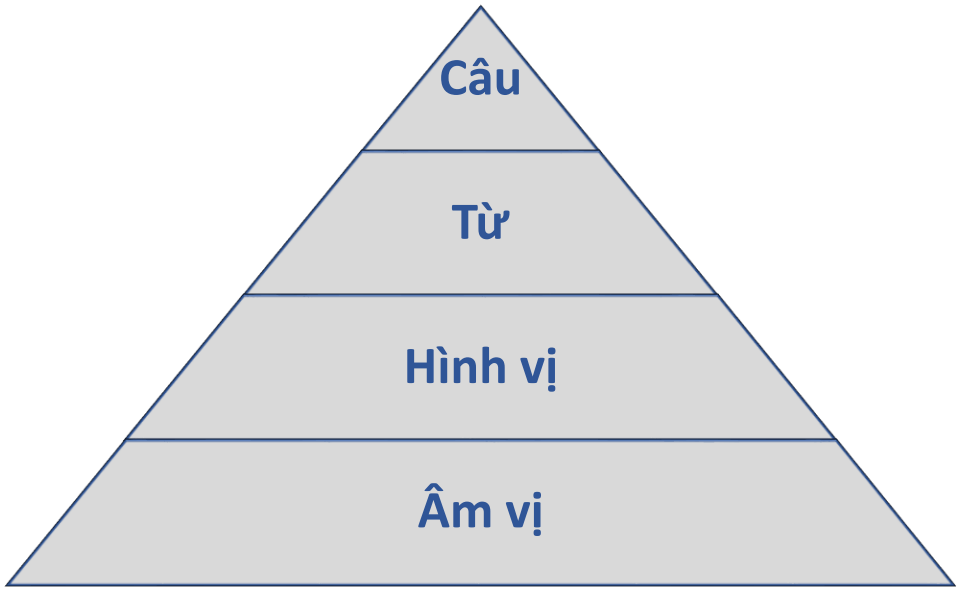
Bản chất của ngôn ngữ tự nhiên:

- Ngôn ngữ là một hiện tượng xã hội: Ngôn ngữ chỉ sinh ra và phát triển trong xã hội loài người. Bên ngoài xã hội, ngôn ngữ không thể phát sinh. Ngôn ngữ không phải là hiện tượng của cá nhân tôi hay cá nhân anh mà là của chúng ta. Đối với mỗi cá nhân, ngôn ngữ như một thiết chế xã hội chặt chẽ, được giữ gìn và phát triển trong kinh nghiệm, trong truyền thống chung của cả cộng đồng.
- Ngôn ngữ là hiện thực trực tiếp của tư tưởng: Ngôn ngữ là phương tiện để giao tiếp và là công cụ của tư duy. Ngôn ngữ (*langue*) được thực tại hoá trong lời nói (*parole*); và lời nói chính là ngôn ngữ đang hành chức, đang được dùng để giao tiếp giữa người với người.

Các đơn vị của ngôn ngữ tự nhiên:

- Âm vị: Là đơn vị ngữ âm nhỏ nhất mà người ta có thể phân ra được trong chuỗi lời nói, hoàn toàn không thể chia nhỏ chúng ra hơn nữa. âm vị có chức năng nhận cảm và chức năng phân biệt nghĩa. Ví dụ: Các âm b, t, v…
- Hình vị: Là một hoặc chuỗi kết hợp một vài âm vị, biểu thị một khái niệm. Nó là đơn vị nhỏ nhất có ý nghĩa. Chức năng của hình vị là chức năng ngữ nghĩa. Ví dụ, kết hợp "quốc gia" trong tiếng Việt gồm hai hình vị: "quốc" là nước, "gia" là nhà.
- Từ: Là chuỗi kết hợp của một hoặc một vài hình vị mang chức năng gọi tên và chức năng ngữ nghĩa. Ví dụ: Các từ "tủ", "ghế", "đi", "cười"…

- Câu: Là chuỗi kết hợp của một hay nhiều từ, chức năng của nó là chức năng thông báo. Ví dụ: “Tôi đang làm đồ án môn xử lý ngôn ngữ tự nhiên”.



Các đơn vị của ngôn ngữ

Các quan hệ trong ngôn ngữ tự nhiên:

- Quan hệ cấp bậc (Còn gọi quan hệ tôn ti / bao hàm): Là quan hệ giữa các đơn vị ngôn ngữ thuộc các cấp độ khác nhau. Quan hệ này thể hiện ở chỗ: các đơn vị thuộc cấp độ cao bao hàm các đơn vị thuộc cấp độ thấp hơn. Ngược lại, các đơn vị thuộc cấp độ thấp nằm trong đơn vị thuộc cấp độ cao hơn và là thành tố để cấu tạo đơn vị ở cấp độ cao hơn nó.
- Quan hệ ngữ đoạn (quan hệ ngang hay quan hệ tuyến tính): Là quan hệ nối kết các đơn vị thành chuỗi khi ngôn ngữ đi vào hoạt động. Quan hệ này được dựa trên tính hình tuyến của ngôn ngữ: Tính chất này bắt buộc các yếu tố ngôn ngữ phải nối tiếp nhau lần lượt trong dòng lời nói để tạo ra các kết hợp gọi là ngữ đoạn. Ví dụ: Những quyển sách này rất hay, đang ăn cơm …

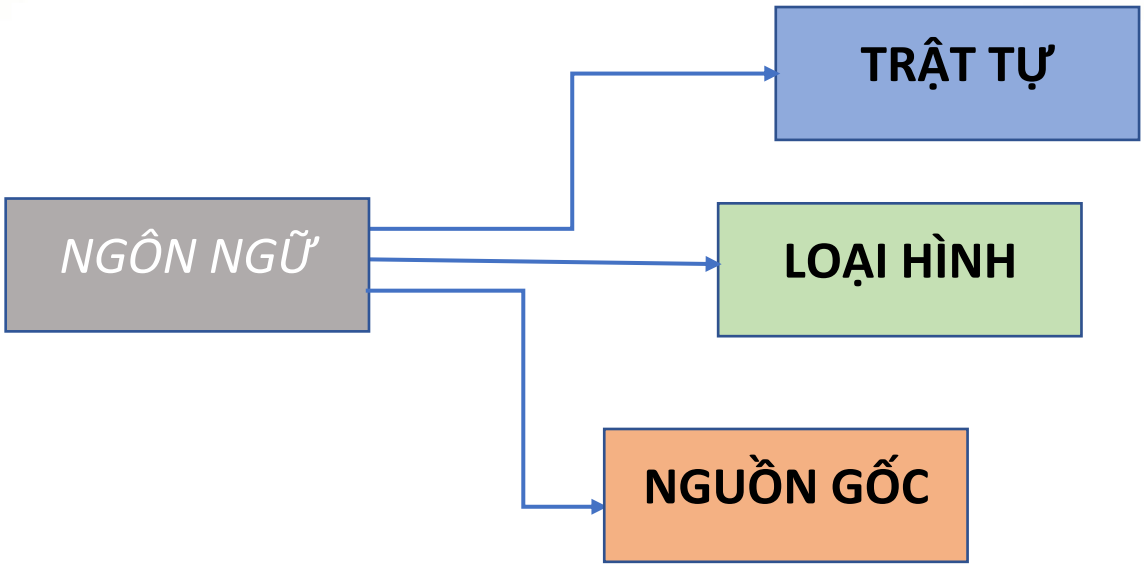
- Quan hệ đối vị (Còn gọi là quan hệ dọc, quan hệ hệ hình): là quan hệ "xâu chuỗi" một yếu tố xuất hiện với những yếu tố vắng mặt "đứng sau lưng nó" và về nguyên tắc có thể thay thế cho nó. Ví dụ: đứng sau lưng từ "trà" trong ngữ đoạn"đang uống trà là một loạt từ như: bia, rượu, cà phê, thuốc, nước …

Các phương diện trong ngôn ngữ tự nhiên:

- Hình thái học: phương thức cấu tạo từ (bằng phương thức phụ tố, căn tố, ghép), phương thức biểu thị các phạm trù ngữ pháp, các ý nghĩa ngữ pháp.
- Cú pháp học: phương thức đánh dấu các thành phần câu, các chức vụ cú pháp, trật tự từ, kết cấu cú pháp.
- Ngữ âm học: thanh điệu, phụ âm, nguyên âm.

Phân loại ngôn ngữ:

- Theo nguồn gốc: Các ngôn ngữ trên thế giới được chia ra khoảng 20 họ khác nhau như họ Nam Á, họ Altai, họ Dravidian, họ Ấn – Âu, họ Hán – Tạng, họ Nam Đảo, họ Thái – Kađai, …
- Theo loại hình: Ngôn ngữ đơn lập (isolate), Ngôn ngữ chắp dính (agglutinate), Ngôn ngữ hòa kết (flexional), ngôn ngữ đa tổng hợp (polysynthetic)
- Theo trật tự từ: Ngôn ngữ chủ-động-tân, ngôn ngữ chủ-tân-động, ngôn ngữ động-chủ-tân, ngôn ngữ V2



Các cách phân loại ngôn ngữ

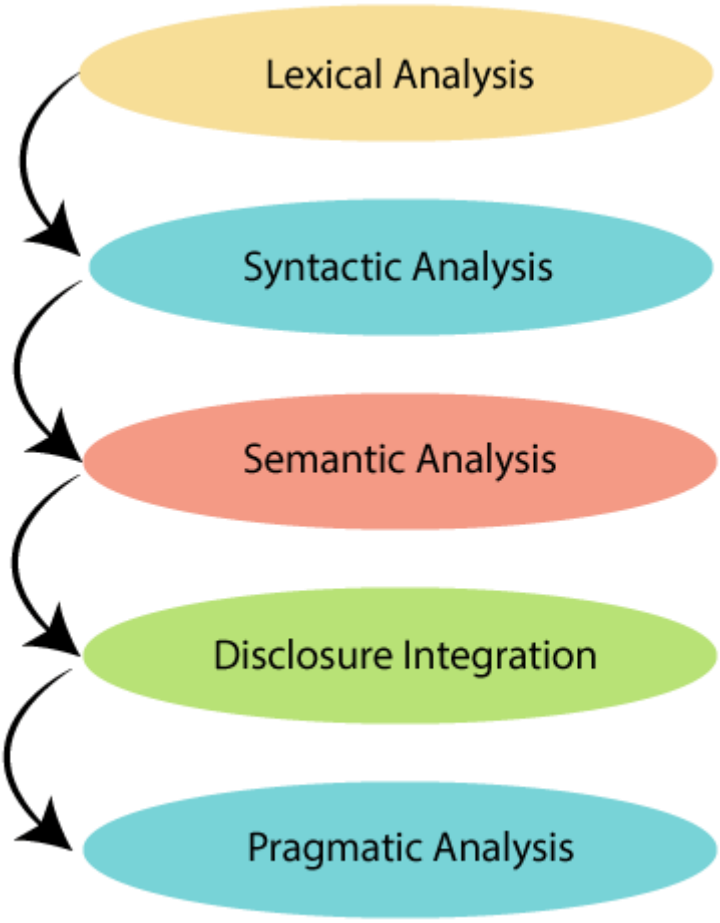
Quá trình xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên gồm 5 giai đoạn: Phân tích từ vựng (lexical analysis), Phân tích cú pháp (syntactic analysis hoặc parsing), Phân tích ngữ nghĩa (semantic analysis), Phân tích diễn ngôn (discourse integration), Phân tích ngữ dụng (pragmatic analysis):

- Phân tích từ vựng (lexical analysis): Đây là bước đầu tiên trong xử lý ngôn ngữ tự nhiên, bao gồm việc định dạng và phân tích cấu trúc của từ vựng. Trong một loại ngôn ngữ, từ vựng bao gồm một tập hợp những từ, ngữ, cụm từ và mệnh đề mang ý nghĩa nhất định. Quá trình này sẽ tra chuỗi các kí tự rồi chuyển đổi chúng thành từ vị chính xác.Phân tích từ vựng chia nhỏ văn bản thành các đoạn văn, câu và từ.
- Phân tích cú pháp (syntactic analysis | parsing): Là quá trình kiểm tra ngữ pháp, vị trí của từ và quan hệ của các từ với nhau, thể hiện mặt logic của các câu trong văn bản hoặc là một bộ phận của những cây ấy. Những luật ngữ

pháp sẽ được áp dụng theo từng trường hợp để thể hiện chính xác nhất cấu trúc của câu.

- Phân tích ngữ nghĩa (semantic analysis): Là quá trình liên hệ cấu trúc ngữ nghĩa, từ cấp độ cụm từ, mệnh đề, câu và đoạn đến cấp độ toàn bài viết, với ý nghĩa độc lập của chúng. Nói cách khác, việc này nhằm tìm ra ngữ nghĩa của đầu vào ngôn từ để cho máy tính có thể hiểu và diễn tả lại câu, đoạn văn hay toàn bộ tài liệu.
- Phân tích diễn ngôn (discourse integration): Là phân tích văn bản có xét tới mối quan hệ giữa ngôn ngữ và ngữ cảnh sử dụng (context-of-use) Quá trình này tập trung vào ngữ nghĩa của cả đoạn văn bằng cách liên hệ các thành phần giữa những câu với nhau. Nghĩa của từ trong một câu có thể phụ thuộc vào ý nghĩa của câu trước hoặc làm tiền đề cho câu sau.
- Phân tích ngữ dụng (pragmatic analysis): Là bước cuối cùng của xử lý ngôn ngữ tự nhiên, là quá trình chiết xuất thông tin, hiểu được ý nghĩa trừu tượng hoặc hàm ý ẩn của câu, từ và văn bản trong các ngữ cảnh nhất định. Phân tích ngữ dụng đòi hỏi kiến thức thực tế, bao gồm ý định, kế hoạch và mục tiêu của người nói, người viết.



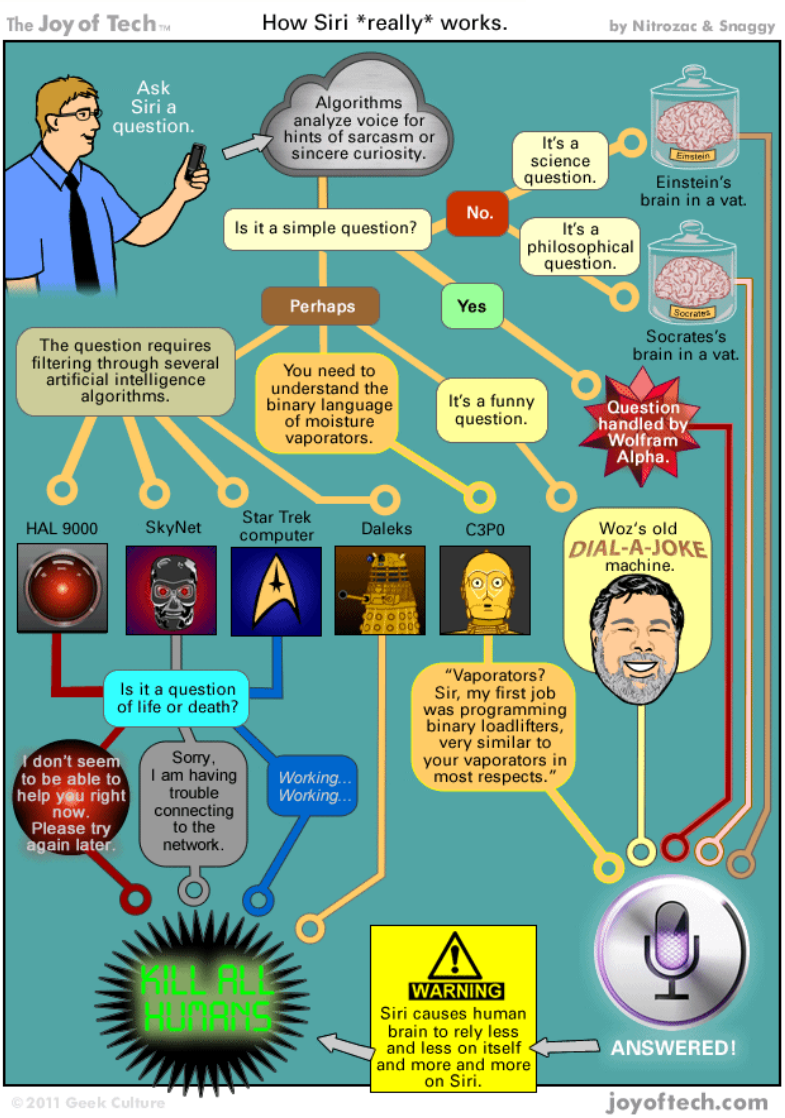
5 bước trong xử lý ngôn ngữ tự nhiên

Một số ứng dụng của xử lý ngôn ngữ tự nhiên

Ngôn ngữ loài người luôn chứa đựng những trường hợp nhập nhằng (ambiguities) khiến việc xác định chính xác ý nghĩa của dữ liệu văn bản hoặc giọng nói là vô cùng khó khăn. Từ đồng âm, châm biếm, thành ngữ, ẩn dụ, ngữ pháp và ngoại lệ sử dụng, các biến thể trong cấu trúc câu. Đây chỉ là một vài điểm bất thường của ngôn ngữ loài người mà con người phải mất nhiều năm để học. Do đó có nhiều ứng dụng xử lý ngôn ngữ tự nhiên ra đời để giải quyết những vấn đề này như: nhận dạng giọng nói (speech recognition), loại bỏ nhập nhằng (word sense disambiguation), lọc thư điện tử (email filters), công cụ tìm kiếm (search engines),… Các ứng dụng này rất thường xuất

hiện bên cạnh chúng ta và hỗ trợ cho chúng ta trong rất nhiều lĩnh vực khác nhau của cuộc sống. Trong số đó có thể kể đến một số ứng dụng nổi bật mà chúng ta thường gặp trong cuộc sống như sau:

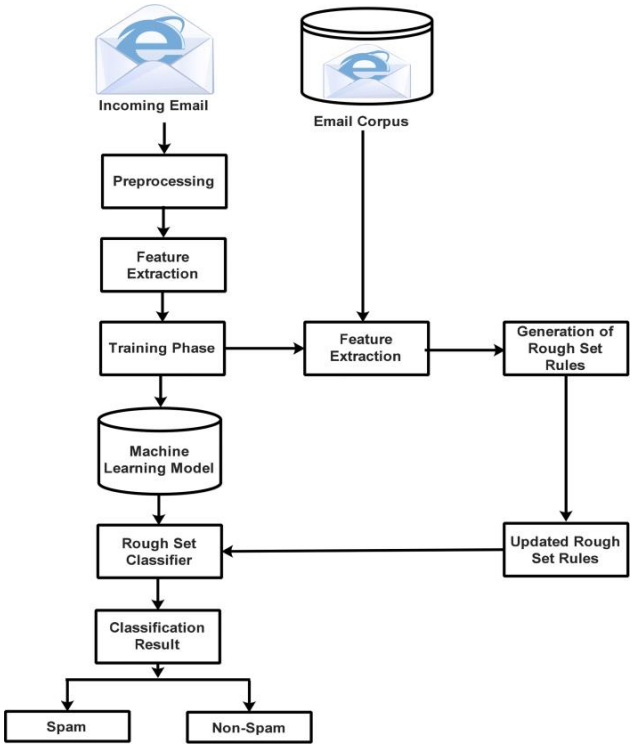
- Nhận dạng giọng nói (speech recognition): Nhận dạng giọng nói được áp dụng đối với bất kỳ ứng dụng nào thực hiện lệnh thoại hoặc trả lời các câu hỏi bằng giọng nói. Điều làm cho việc nhận dạng giọng nói trở nên đặc biệt khó khăn là cách mọi người nói chuyện — nói nhanh, nói lý với sự nhấn mạnh và ngữ điệu khác nhau, ở các trọng âm khác nhau và thường sử dụng ngữ pháp không chính xác. Các trợ lý thông minh như Siri của Apple và Alexa của Amazon nhờ nhận dạng giọng nói mà suy ra ý nghĩa và đưa ra phản hồi hữu ích. Việc Siri hoặc Alexa xuất hiện trong nhà và cuộc sống hàng ngày của chúng ta khi chúng ta trò chuyện thông qua các bộ điều nhiệt, công tắc đèn, ô tô, điện thoại thông minh,… cũng đang hiện hữu dần. Nhờ vậy, chất lượng cuộc sống được nâng cao và những hoạt động thường ngày diễn ra dễ dàng hơn.



Cách thức mà Siri hoạt động

- Lọc thư điện tử (email filters): Bộ lọc email là một trong những ứng dụng cơ bản nhất của NLP, bắt đầu với các bộ lọc thư rác, phát hiện ra các từ hoặc cụm từ nhất định báo hiệu một tin nhắn rác. Nhưng tính năng lọc đã được nâng cấp, giống như các bản thích ứng ban đầu của NLP. Một trong những ứng dụng phổ biến hơn, mới hơn của NLP được tìm thấy trong phân loại email của Gmail. Hệ thống nhận biết nếu email thuộc một trong ba danh mục (chính, xã hội hoặc quảng cáo) dựa trên nội dung của chúng. Đối với tất cả người dùng Gmail, điều này giúp hộp thư đến dễ dàng quản lí hơn với

các email quan trọng, có liên quan mà người dùng muốn xem lại và trả lời nhanh chóng.



Quá trình phân loại thư spam

- Công cụ tìm kiếm (search engines): Các công cụ tìm kiếm sử dụng NLP để hiển thị các kết quả có liên quan dựa trên các hành vi tìm kiếm tương tự hoặc mục đích của người dùng để tìm thấy những gì họ cần. Ví dụ: Google không chỉ dự đoán những tìm kiếm phổ biến nào có thể áp dụng cho truy vấn khi thanh tìm kiếm được nhập, mà còn xem xét tổng thể và nhận ra ý muốn người dùng thay vì chỉ dựa trên những từ được nhập vào. Ai đó có thể nhập số hiệu chuyến bay vào Google và nhận trạng thái chuyến bay, nhập ký hiệu đánh dấu và nhận thông tin chứng khoán hoặc máy tính có thể xuất hiện khi nhập một phương trình toán học. Đây là một số ví dụ khi hoàn thành tìm kiếm vì xử lý ngôn ngữ tự nhiên trong tìm kiếm liên kết truy vấn nhập nhằng với các thực thể tương đối và cung cấp kết quả hữu ích.

Một số công cụ tìm kiếm hiện nay

Những thách thức của xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên tuy là một nhánh trong lĩnh vực trí tuệ nhân tạo mới phát triển gần đây nhưng đã là công cụ mạnh mẽ với những lợi ích to lớn. Tuy nhiên, vẫn còn một số hạn chế và vấn đề trong xử lý ngôn ngữ tự nhiên:

- Từ, cụm từ theo ngữ cảnh và từ đồng âm: Các từ, cụm từ giống nhau có thể có nghĩa khác nhau tùy theo ngữ cảnh của một câu và nhiều từ có cách phát âm giống hệt nhau nhưng nghĩa hoàn toàn khác nhau. Từ đồng âm - hai hoặc nhiều từ được phát âm giống nhau nhưng có định nghĩa khác nhau - có thể gây khó khăn cho các ứng dụng trả lời câu hỏi và chuyển lời nói thành văn bản vì chúng không được viết ở dạng văn bản. VD: từ "kho" trong *kho cá* và *nhà kho* mang ý nghĩa khác nhau trong hai trường hợp.
- Từ đồng nghĩa: Từ đồng nghĩa có thể dẫn đến các vấn đề trong ngữ cảnh từ nếu sử dụng nhiều từ khác nhau để diễn đạt cùng một ý tưởng. Hơn nữa, một số từ trong số này có thể truyền đạt cùng một ý nghĩa, trong khi một số chỉ dùng để thể hiện mức độ. VD: từ *chết* và *hy sinh* đều có chung một nghĩa

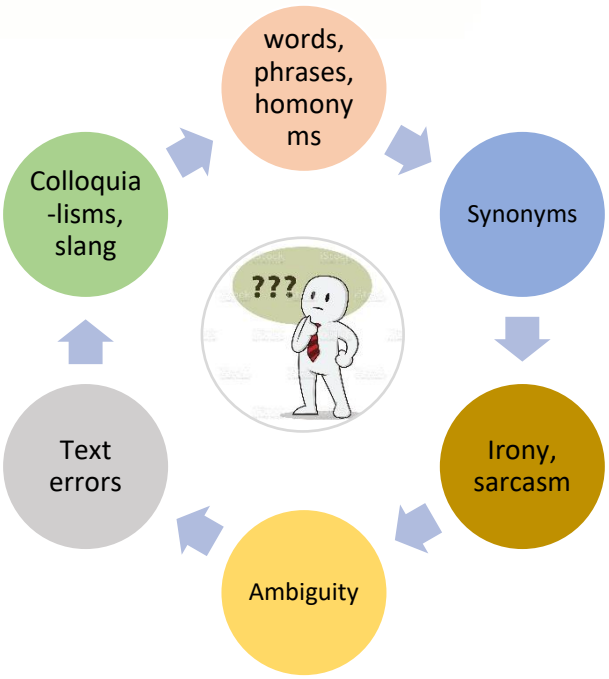
nhưng lại thể hiện hai mức độ tôn trọng khác nhau nếu dùng chung trong câu. Còn có trường hợp sử dụng từ đồng nghĩa để biểu thị các ý nghĩa khác nhau trong vốn từ vựng cá nhân. Vì vậy, để xây dựng hệ thống NLP, điều quan trọng là phải bao gồm tất cả các nghĩa có thể có của một từ và tất cả các từ đồng nghĩa có thể có. Các mô hình phân tích văn bản đôi khi vẫn có thể mắc lỗi, nhưng càng nhận được nhiều dữ liệu đào tạo phù hợp, mô hình có thể hiểu các từ đồng nghĩa tốt hơn.

- Mỉa mai và châm biếm: Sự mỉa mai và châm biếm là một vấn đề không nhỏ đối với các mô hình học máy thuộc xử lý ngôn ngữ tự nhiên vì chúng thường sử dụng các từ và cụm từ, theo đúng định nghĩa sẽ mang hàm ý tích cực hoặc tiêu cực, nhưng thực tế lại có ý nghĩa ngược lại.
- Nhập nhằng: Tính nhập nhằng trong xử lý ngôn ngữ tự nhiên đề cập đến các câu và cụm từ có khả năng có hai hoặc nhiều cách hiểu. Bao gồm: sự nhập nhằng về từ ngữ - một từ có thể được sử dụng như một động từ, danh từ hoặc tính từ, sự nhập nhằng về ngữ nghĩa - việc giải thích một câu trong ngữ cảnh. Ví dụ: *Ông già đi nhanh quá*. Điều này có thể có nghĩa là một cụ già di chuyển nhanh hay một người bảo người khác già nhanh quá, sự nhập nhằng về cú pháp - là một tình huống trong đó một câu có thể được hiểu theo nhiều cách do cấu trúc câu không rõ ràng.
- Lỗi trong văn bản và giọng nói: Các từ viết sai chính tả hoặc sử dụng sai có thể tạo ra các vấn đề cho việc phân tích văn bản. Các ứng dụng tự động sửa lỗi và sửa ngữ pháp có thể xử lý các lỗi phổ biến, nhưng không phải lúc nào cũng hiểu được ý định của người viết. Với ngôn ngữ nói, việc phát âm sai, nhấn giọng khác nhau, nói lắp,… có thể gây khó hiểu đối với máy móc.
- Từ thông tục và tiếng lóng: Các cụm từ thông tục, cách diễn đạt, thành ngữ và biệt ngữ dành riêng cho từng văn hóa đặt ra một số vấn đề đối với xử lý ngôn ngữ tự nhiên. Đặc biệt là đối với các mô hình dành cho mục đích sử dụng rộng rãi. Bởi vì trong ngôn ngữ trang trọng, các từ thông tục có thể không có "định nghĩa từ điển" nào và những cách diễn đạt của chúng thậm chí có thể có nghĩa khác nhau ở các khu vực địa lý khác nhau. Hơn nữa, tiếng lóng trong văn hóa không ngừng biến đổi và mở rộng, vì vậy những từ mới

xuất hiện mỗi ngày. Việc đào tạo và cập nhật thường xuyên các mô hình tùy chỉnh có thể hữu ích, mặc dù đôi khi việc đó đòi hỏi khá nhiều ngữ liệu đầu vào.

- Ngôn ngữ dành riêng cho từng lĩnh vực: Các doanh nghiệp và ngành công nghiệp khác nhau thường sử dụng ngôn ngữ khác nhau. Ví dụ, một mô hình xử lý ngôn ngữ tự nhiên trong ngành y sẽ rất khác so với một mô hình được sử dụng để xử lý các văn bản pháp lý. Tuy nhiên, ngày nay, có một số công cụ phân tích được đào tạo cho các lĩnh vực cụ thể, nhưng các ngành cực kỳ thích hợp có thể cần xây dựng hoặc đào tạo mô hình của riêng họ.
- Ngôn ngữ ít tài nguyên: Nhiều ngôn ngữ, đặc biệt là những ngôn ngữ được nói bởi những người ít tiếp cận với công nghệ thường bị bỏ qua và chưa được xử lý. Ví dụ, theo một số ước tính, (tùy thuộc vào ngôn ngữ so với phương ngữ), chỉ riêng ở châu Phi đã có hơn 3.000 ngôn ngữ. Đơn giản vì không có nhiều dữ liệu về những ngôn ngữ này.

Mặc dù xử lý ngôn ngữ tự nhiên có những hạn chế nhất định nhưng nó vẫn mang lại những lợi ích to lớn trên phạm vi rộng cho bất kỳ doanh nghiệp nào. Với những kỹ thuật mới và công nghệ mới được tạo nên hàng ngày, nhiều rào cản trong số này sẽ bị phá vỡ trong những năm tới. Máy học trong xử lý ngôn ngữ tự nhiên có thể hoạt động để phân tích lượng lớn văn bản trong thời gian thực tế để có được những thông tin chi tiết chưa từng có trước đây.



Các khó khăn trong xử lý ngôn ngữ tự nhiên

2. Sơ lược gán nhãn từ loại

Khái quát

Trong thực tế, việc xác định từ loại trong câu luôn là một vấn đề cơ bản mà mỗi cá nhân đã được tiếp xúc ngay tại trương trình tiểu học. Ví dụ, nếu cho câu "Cô ấy rất thích đồ ngọt", ta có thể dễ dàng xác định được các thành phần trong câu như thế này:

- Đại từ: Cô ấy.
- Phó từ: rất
- Động từ: thích
- Danh từ: đồ ngọt

Trong NLP, việc xác định từ loại trong câu hay gán nhãn từ loại (POS tagging) được áp dụng trong các bài toán như gán nhãn tên thực thể (named Entity Recognition), phân tích cảm xúc (sentiment analysis), trả lời nghi vấn (question answering), loại bỏ nhập nhằng (word sense disambiguation).

Do đó, một trong các vấn đề nền tảng của ngôn ngữ tự nhiên là việc phân loại các từ thành các lớp từ loại dựa theo thực tiễn hoạt động ngôn ngữ. Mỗi từ loại tương ứng với một lớp từ giữ một vai trò ngữ pháp nhất định. Nói chung, mỗi từ trong một ngôn ngữ có thể gắn với nhiều từ loại, và việc tự động "hiểu" đúng nghĩa một từ phụ thuộc vào việc nó được xác định đúng từ loại hay không. Công việc gán nhãn từ loại cho một văn bản là xác định từ loại của mỗi từ trong phạm vi văn bản đó. Các công cụ gán nhãn (hay chú thích) từ loại cho các từ trong một văn bản có thể thay đổi tuỳ theo quan niệm về đơn vị từ vựng và thông tin ngôn ngữ cần khai thác trong các ứng dụng cụ thể.

Xác định từ loại chính xác cho các từ trong văn bản là vấn đề rất quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên. Công cụ gán nhãn từ loại có thể được ứng dụng rộng rãi trong các hệ thống tìm kiếm thông tin, trong các ứng dụng tổng hợp tiếng nói, các hệ thống nhận dạng tiếng nói cũng như trong các hệ thống dịch máy. Công cụ này cũng hỗ trợ cho việc phân tích cú pháp các văn bản, góp phần giải quyết tính đa nghĩa của từ, và trợ giúp các hệ thống truy hồi thông tin hướng đến ngữ nghĩa, …

Gán nhãn từ loại có thể được làm thủ công. Tuy nhiên, trong ngữ cảnh ngôn ngữ học tính toán, khi ngữ liệu đầu vào nhiều thì việc gán nhãn từ loại sẽ được thực hiện tự động bằng máy tính, sử dụng các thuật toán liên kết với các thuật ngữ rời rạc, cũng như các trạng thái ẩn của ngữ liệu, bằng một tập hợp các nhãn cho trước.

Nhãn từ loại (tagset)

Thông thường, ta chỉ biết các thành phần câu đơn giản như: danh từ, động từ, mạo từ, tính từ, giới từ, đại từ, trạng từ, liên từ và thán từ. Tuy vậy, trên thực tế, một từ có thể chia thành nhiều danh mục và tiểu mục hơn. Đối với danh từ, có thể phân biệt dạng số nhiều, dạng sở hữu và dạng số ít. Trong nhiều ngôn ngữ, các từ cũng được đánh dấu theo "trường hợp" (vai trò làm chủ ngữ, tân ngữ,…), giới tính ngữ pháp,…; các động từ được đánh dấu theo thì, khía cạnh, và những thứ khác. Trong một số hệ thống gắn thẻ, các cách sử dụng khác nhau của cùng một từ gốc sẽ được gán các nhãn khác nhau của lời nói, dẫn đến một số lượng lớn các thẻ. Ví dụ: NN cho danh từ chung số ít, NNS cho danh từ chung số nhiều, NP cho danh từ riêng số ít (bộ nhãn Brown Corpus).

Các hệ thống gắn thẻ khác sử dụng số lượng thẻ ít hơn và bỏ qua những điểm khác biệt nhỏ hoặc mô hình hóa chúng thành các tính năng phần nào độc lập với các nhãn từ loại.

Bộ nhãn phổ biến nhất cho tiếng Anh – Mỹ có lẽ là bộ thẻ Penn, được phát triển trong dự án Penn Treebank. Nó gần giống với các bộ thẻ Brown Corpus và LOB Corpus, mặc dù nhỏ hơn nhiều. Ở Châu Âu, các bộ nhãn từ Eagles Guidelines được sử dụng rộng rãi và bao gồm các phiên bản khác nhau cho nhiều ngôn ngữ. Ngoài ra, còn có bộ nhãn VLSP sử dụng các nhãn từ loại theo dự án nghiên cứu cùng tên để gán nhãn cho hơn 10000 câu tiếng Việt.

Tag	Description	Example	Tag	Description	Example
CC	coord. conjunction	*and, or*	RB	adverb	*extremely*
CD	cardinal number	*one, two*	RBR	adverb, comparative	*never*
DT	determiner	*a, the*	RBS	adverb, superlative	*fastest*
EX	existential there	*there*	RP	particle	*up, off*
FW	foreign word	*noire*	SYM	symbol	*+, %*
IN	preposition or sub-conjunction	*of, in*	TO	"to"	*to*
JJ	adjective	*small*	UH	interjection	*oops, oh*
JJR	adject., comparative	*smaller*	VB	verb, base form	*fly*
JJS	adject., superlative	*smallest*	VBD	verb, past tense	*flew*
LS	list item marker	*1, one*	VBG	verb, gerund	*flying*
MD	modal	*can, could*	VBN	verb, past participle	*flown*
NN	noun, singular or mass	*dog*	VBP	verb, non-3sg pres	*fly*
NNS	noun, plural	*dogs*	VBZ	verb, 3sg pres	*flies*
NNP	proper noun, sing.	*London*	WDT	wh-determiner	*which, that*
NNPS	proper noun, plural	*Azores*	WP	wh-pronoun	*who, what*
PDT	predeterminer	*both, lot of*	WP$	possessive wh-	*whose*
POS	possessive ending	*'s*	WRB	wh-adverb	*where, how*
PRP	personal pronoun	*he, she*			

Bộ nhãn Penn Treebank

Phương pháp

Gán nhãn từ loại có thể được chia thành 2 nhóm:

- Gán nhãn dựa trên luật (Ruled – based tagging)

- Gán nhãn thống kê (Stochastic tagging)

Gán nhãn dựa trên luật: Các mô hình gán nhãn dựa trên luật áp dụng các quy tắc dành cho văn bản viết tay và sử dụng thông tin ngữ cảnh để gán nhãn cho các từ. Các quy tắc này thường được xem như quy chuẩn. Ví dụ: "Trong tiếng Anh, nếu một từ X không xác định đứng trước một mạo từ và theo sau là một danh từ, X sẽ được xem như tính từ". Việc phân loại được thực hiện bằng cách phân tích các đặc điểm ngôn ngữ của từ, từ đứng trước, từ sau và các khía cạnh khác. Gán nhãn dựa trên luật là phương pháp mang lại độ chính xác cao và dễ thực hiện. Tuy nhiên, việc xác định một bộ luật theo cách thủ công là một quá trình cực kỳ phức tạp và hoàn toàn không thể mở rộng. Vì vậy, trong xử lý ngôn ngữ tự nhiên, các phương pháp thống kê sẽ thường được sử dụng nhiều hơn.

Gán nhãn thống kê: Phương pháp gán nhãn thống kê bao gồm việc áp dụng các khái niệm toán học như tần suất, xác suất hoặc thống kê vào mô hình. Cách tiếp cận đơn giản nhất là gán nhãn từ chỉ dựa trên xác suất từ đó xuất hiện với các nhãn cụ thể. Nói cách khác, nhãn gặp phải thường xuyên nhất trong tập huấn luyện với từ tương ứng là nhãn được gán cho trường hợp nhập nhằng của từ đó. Vấn đề với cách tiếp cận này là mặc dù nó có thể mang lại một nhãn hợp lệ cho một từ nhất định, nhưng nó cũng có thể mang lại chuỗi nhãn sai lệch với kết quả mong muốn. Mức độ cao hơn của gán nhãn thống kê kết hợp hai cách tiếp cận trước đó, sử dụng cả xác suất chuỗi nhãn và phép đo tần suất từ. Đây được gọi là Mô hình Markov ẩn (HMM) – phương pháp mà nhóm chúng tôi sẽ sử dụng trong báo cáo này

3. Phát biểu bài toán

Đầu vào (Input)

Đối tượng gán nhãn từ loại là ngôn ngữ viết. Đầu vào có thể là một câu, một đoạn văn hay cả văn bản đã được tách từ và bộ nhãn(tagset) tương ứng như: Bộ nhãn từ loại Penn Treebank, Brown corpus, Universal Dependencies hoặc VLSP cho tiếng Việt.

Đầu ra (Output)

Văn bản đã trải qua tiền xử lý là tách từ với từng từ được gán với nhãn tương ứng dựa trên bộ nhãn được chọn.

Ví dụ minh họa

Để minh họa một cách trực quan mô hình của bài toán, chúng tôi xin đưa ra một ví dụ như sau.

Cho trước hai câu: "kho chứa cá thật tanh" và "cô ấy kho cá" thì ta có kết quả gán nhãn là:

- Câu 1: kho/N chứa/V cá/N thật/A tanh/A
- Câu 2: Cô/N ấy/P kho/V cá/N

(Sử dụng bộ nhãn của từ điển VLSP)

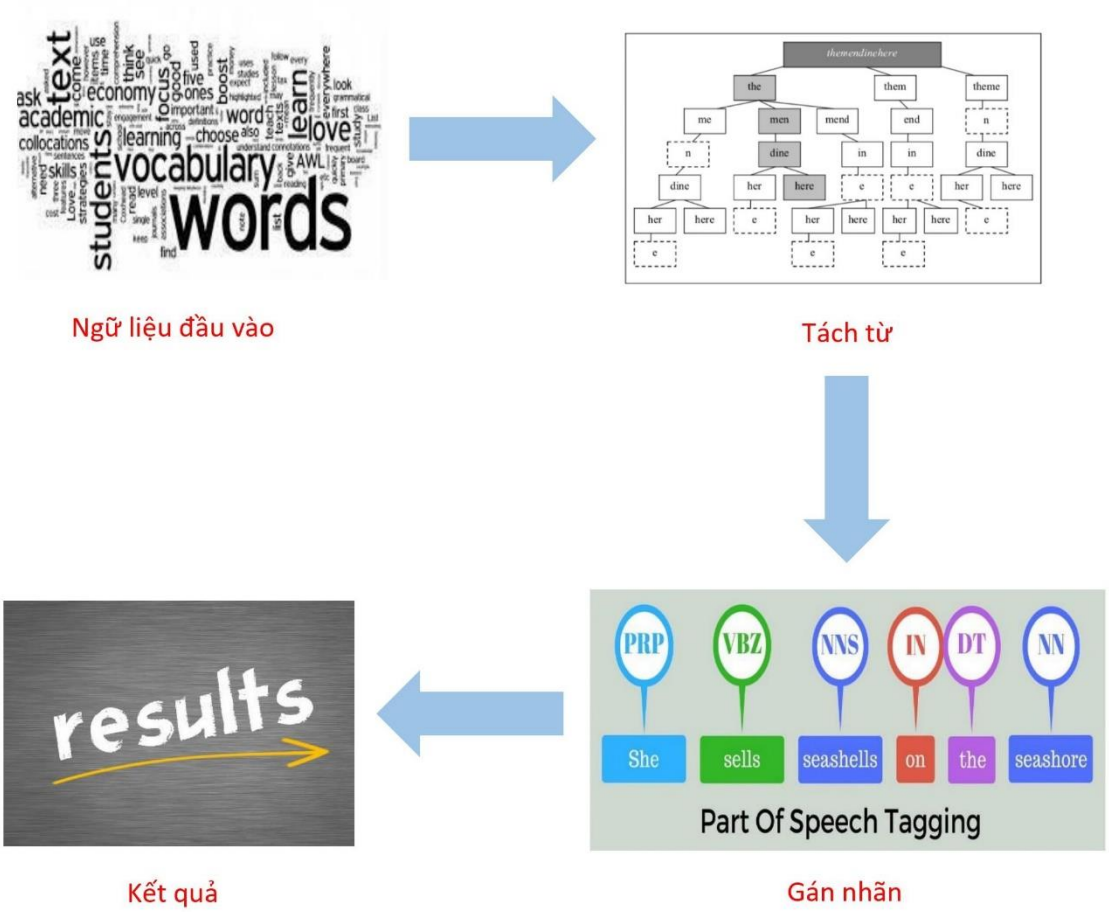
Ở hai câu, chữ "kho" mang hai nhãn khác nhau theo từng trường hợp. Trong câu 1, "kho" có từ loại là danh từ, mang nghĩa "nhà kho" – một tòa nhà được xây dựng trên một địa điểm đạt các điều kiện nhất định để sử dụng cho việc chứa và lưu trữ của cải, sản phẩm, hàng hóa hoặc nguyên vật liệu. Còn từ "kho" trong câu 2 là một động từ chỉ hành động dùng nhiệt nấu kỹ thức ăn cho ngấm gia vị theo hướng mặn.

Có thể thấy, đây là một trong những trường hợp nhập nhằng điển hình trong xử lý ngôn ngữ tự nhiên. Việc gán nhãn từ loại sẽ loại bỏ đi sự nhập nhằng của câu, là tiền đề của các bài toán bậc cao hơn.

4. Mô hình tổng quát

Trong báo cáo này, mô hình gán nhãn từ loại sẽ được xây dựng dựa trên mô hình Markov ẩn (Hidden Markov Model) như đã nhắc đến ngay trong tên của tập báo cáo này với khâu tiền xử lý là sử dụng so khớp dài nhất (Longest Matching) để tách từ cho ngữ liệu đầu vào. Các phương pháp đánh giá độ chính xác của mô hình cả về phần tách từ lẫn gán nhãn sẽ được trình bày trong phần sau của báo cáo này. Với mô hình này, các ngữ liệu đầu vào sẽ trải qua quá trình tách từ rồi kết quả tách từ sẽ được sử dụng cho việc gán nhãn. Sau khi hệ thống xử lý xong, kết quả trả về sẽ là chuỗi nhãn tương ứng.

Bài toán có thể được phát biểu một cách trực quan và ngắn gọn dưới dạng mô hình như sau:



Mô hình tổng quát của bài toán

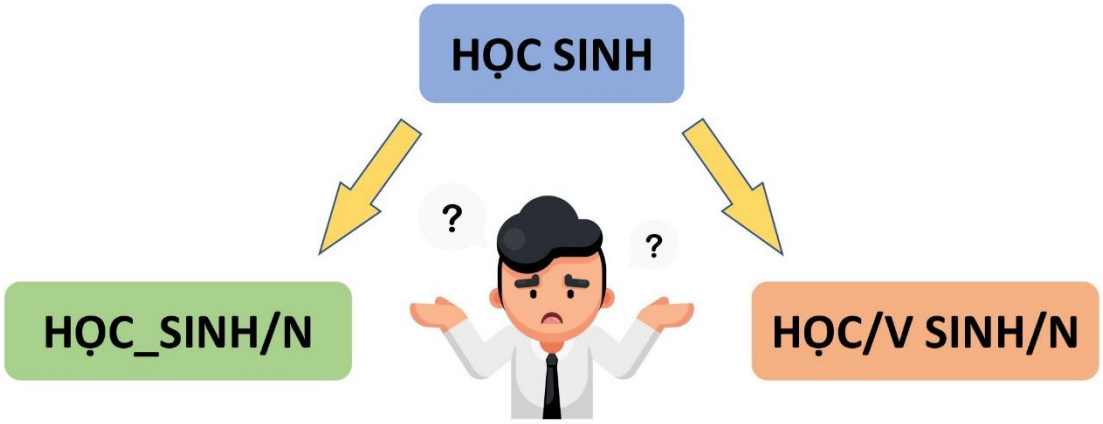
5. Các thách thức của bài toán

Việc xây dựng một mô hình có khả năng giải quyết yêu cầu của bài toán đặt ra tuy có nhiều thuận lợi nhờ vào sự phát triển của máy tính hiện đại, ngôn ngữ lập trình và các công cụ lập trình. Tuy nhiên, theo nhóm nhận thấy, việc xây dựng mô hình này vẫn còn gặp phải một số thách thức trong quá trình thực hiện như "Trường hợp gán nhãn từ ghép", "Việc có sai sót ở phần tách từ dẫn đến việc gán nhãn không chính xác", "Các trường hợp sai lỗi ngữ pháp, viết tắt ở

phần ngữ liệu đầu vào". Đó cũng chính là những trở ngại mà chúng tôi gặp phải trong quá trình thực hiện đề tài.

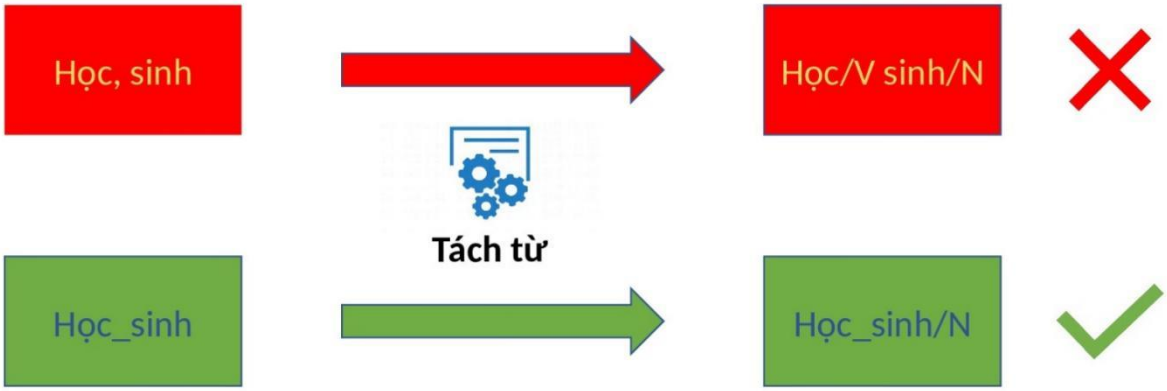
Các thách thức của bài toán

- Trường hợp gán nhãn từ ghép:
  - Khác với những từ trong tiếng anh hầu hết là từ đơn nghĩa, hoặc nếu có ghép hai từ đơn nghĩa với nhau thì cũng không có dấu cách ở giữa các từ, khi thực hiện gán nhãn cho tiếng thì đã có một lượng lớn các từ trong từ điển tiếng Việt là từ ghép, cấu tạo từ hai từ đơn có nghĩa nên khi thực hiện gán nhãn, máy tính có thể gán nhãn nhầm cho từ đơn có trong từ phức. Điều này có thể thấy trong các mô hình gán nhãn tiếng Việt. Kết quả trả về có thể không giống với kết quả mong muốn.
  - Để khắc phục tình trạng này thì nên xử lý từ ghép trước giai đoạn gán nhãn, cụ thể là xử lý từ bước tách từ. Một cách là thay vì cách các từ đơn, một dấu phân cách sẽ được đặt giữa các từ như dấu "_". VD: từ "học tập" sẽ thành "học_tập".



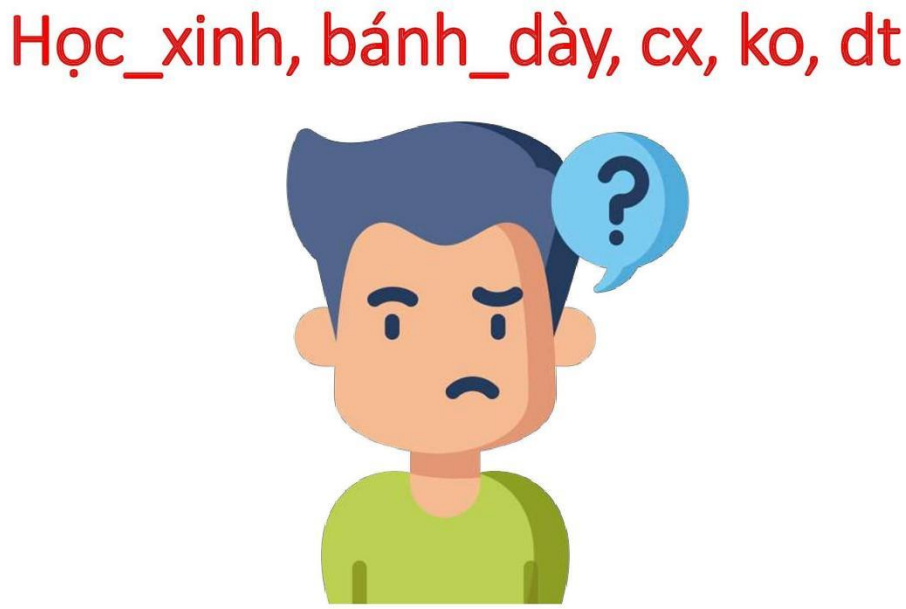
Trường hợp gán nhãn từ ghép

- Việc có sai sót ở phần tách từ dẫn đến việc gán nhãn không chính xác:
  - Ở giai đoạn tách từ, tùy vào phương pháp mà có thể dẫn đến độ chính xác khác nhau. Do trong tiếng Việt, từ không phải là đơn vị nhỏ nhất trong câu mà là tiếng. Một từ được cấu trúc từ các tiếng, có thể từ một tiếng (từ đơn) hoặc bao gồm nhiều tiếng (từ phức) bao gồm cả từ láy và từ ghép nên rất khó có thể áp dụng các kỹ thuật và hướng tiếp cận đã được nghiên cứu và thử nghiệm thành công trên các ngôn ngữ Ấn – Âu cho tiếng Việt. Khi độ chính xác thấp sẽ dẫn đến kết quả tách từ sai nhất là các trường hợp nhầm lẫn từ ghép, từ láy với các tiếng trong chúng. VD: từ "học sinh" có thể tách thành "học, sinh" thay vì"học_sinh".



Tách từ sai dẫn đến gán nhãn sai

- Các trường hợp sai lỗi ngữ pháp, viết tắt ở phần ngữ liệu đầu vào:
  - Đây cũng là một vấn đề ở phần tiền xử lý ngữ liệu, các từ viết tắt hoặc sai chính tả sẽ gây ra khó khăn trong quá trình sản sinh chuỗi nhãn, nhất là đối với các phương pháp gán nhãn thống kê, khi mà nhãn từ loại này có thể phụ thuộc vào thành phần đứng trước.
  - Để giảm thiểu sai sót, ta có thể sửa thủ công từng lỗi chính tả hoặc lỗi ngữ pháp. Tuy nhiên, nếu ngữ liệu đầu vào lớn thì cách này sẽ không khả quan. Cách khắc phục được đề xuất là gán một nhãn mặc định cho các từ không xác định như là danh từ, tính từ hay một nhãn dành riêng cho các từ ấy.



Các trường hợp sai lỗi chính tả, viết tắt

6. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là mô hình tách từ và gán nhãn từ loại cho văn bản tiếng Việt.

Mô hình tách từ được xây dựng dựa trên thuật toán longest matching còn phần gán nhãn sẽ sử dụng mô hình Markov ẩn với các phương pháp smoothing khác nhau.

Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài được giới hạn trong phạm vi các câu tiếng Việt

Cụ thể, nhóm đã xây dựng được bộ ngữ liệu gồm 40 câu tiếng Việt trong suốt quá trình làm báo cáo này với các trường hợp nhập nhằng, các từ không xác định khác nhau để kiểm tra độ chính xác của mô hình trên các tình huống khác nhau

7. Mục tiêu bài toán

Dựa trên cơ sở kiến thức về phương pháp longest matching và mô hình Markov ẩn, đề tài đưa ra định hướng xây dựng một mô hình tách từ và gán nhãn cho các câu tiếng Việt. Từ các câu tiếng Việt được cung cấp trước đó (rời rạc hoặc liên tục). Mô hình được xây dựng trong đề tài này sẽ trả về chuỗi nhãn được tạo ra ứng với các câu từ ngữ liệu đầu vào. Mô hình này cần phải đạt được các yêu cầu như có khả năng tách từ thông qua thuật toán longest matching, có khả năng tạo ra chuỗi nhãn chính xác đúng với kết quả dự tính. Đồng thời, đề tài còn đưa ra kết quả từ việc thử nghiệm mô hình gán nhãn được xây dựng, từ đó có thể đánh giá khách quan mô hình khi xét về chất lượng và độ chính xác của kết quả trả về dựa trên việc đánh giá mô hình thông qua các phương pháp khác nhau như accuracy, precision và recall.

Thông qua mô hình được xây dựng trong đề tài này, nhóm hi vọng rằng sẽ mang lại một công cụ hữu ích cho việc tách từ và gán nhãn, hai bài toán tiền đề cho các vấn đề cao cấp hơn trong xử lý ngôn ngữ tự nhiên

Chương 2. CƠ SỞ LÝ THUYẾT

I. Tách từ

1. Giới thiệu

Tách từ là một quá trình xử lý nhằm mục đích xác định ranh giới của các từ trong câu văn, cũng có thể hiểu đơn giản rằng tách từ là quá trình xác định các từ đơn, từ ghép… có trong câu. Đối với xử lý ngôn ngữ, để có thể xác định cấu trúc ngữ pháp của câu, xác định từ loại của một từ trong câu, yêu cầu nhất thiết đặt ra là phải xác định được đâu là từ trong câu. Vấn đề này tưởng chừng đơn giản với con người nhưng đối với máy tính thì đây là bài toán rất khó giải quyết.

Tách từ được xem là bước xử lý quan trọng đối với các hệ thống xử lý ngôn ngữ tự nhiên, đặc biệt là đối với các ngôn ngữ thuộc vùng Đông Á theo loại hình ngôn ngữ đơn lập, ví dụ: tiếng Trung Quốc, tiếng Nhật, tiếng Thái, và tiếng Việt. Với các ngôn ngữ thuộc loại hình này, ranh giới từ không chỉ đơn giản là những khoảng trắng như trong các ngôn ngữ thuộc loại hình hòa kết như tiếng Anh…, mà có sự liên hệ chặt chẽ giữa các tiếng với nhau, một từ có thể cấu tạo bởi một hoặc nhiều tiếng. Vì vậy đối với các ngôn ngữ thuộc vùng Đông Á, vấn đề của bài toán tách từ là khử được sự nhập nhằng trong ranh giới từ.

Hiện tại đã có rất nhiều công trình nghiên cứu để giải quyết vấn đề này và có độ chính xác khá cao (>95%).

Vấn đề của bài toán tách từ là khử được sự nhập nhằng trong ranh giới từ.

2. Các hướng tiếp cận trong tách từ

Phương pháp tiếp cận dựa trên từ

Tiếp cận dựa vào từ điển cố định:

- Đây là phương pháp điển hình nhất hiện nay, độ chính xác cũng khá cao. Ý tưởng của phương pháp này là dựa vào 1 từ điển từ có sẵn rồi dùng các biện pháp so khớp để tách ra các từ, cụm từ trong văn bản mà có trong từ điển. Các hướng tiếp cận khác nhau sẽ sử dụng các loại từ điển khác nhau: full-

word/pharse sẽ sử dụng một bộ từ điển hoàn chỉnh, trong khi đó component lại sử dụng các bộ từ điển thành phần.

- Phương pháp này cũng được chia làm nhiều loại dựa theo các so khớp bới từ điển, chẳng hạn như so khớp dài nhất (longest matching) hay so khớp ngắn nhất (shortest matching). Ngoài ra còn phương pháp kết hợp (hybird) kết hợp cả hai phương pháp này. Hiện nay thì phương pháp so khớp dài nhất được xem là phương pháp hiệu quả nhất trong hướng tiếp cận này.
- Hạn chế của hướng tiếp cận này là kết quả phụ thuộc hoàn toàn vào độ chính xác và đầy đủ của từ điển. Việc xây dựng một bộ từ điển hoàn chỉnh là cấn đề cốt lõi của hướng tiếp cận này, hiện nay với sự nỗ lực của nhiều người, bộ từ điển đã tương đối đầy đủ đem lại kết quả khá khả quan cho phương pháp này với độ chính xác cao (95%) trong việc tách từ.

Tiếp cận dựa vào thống kê thuần túy:

- Ý tưởng của hướng tiếp cận này là dựa vào các thông tin như tần số xuất hiện trong tập dữ liệu huấn luyện ban đầu, dựa vào các giải thuật học máy sẽ đưa ra một tập các từ được gán trọng số. Dựa trên các trọng số này, khi phân tách câu sẽ quyết định một cụm các tiếng có phải là một từ hay không. Ý tưởng của hướng tiếp cận này là dựa vào các thông tin như tần số xuất hiện trong tập dữ liệu huấn luyện ban đầu, dựa vào các giải thuật học máy sẽ đưa ra một tập các từ được gán trọng số. Dựa trên các trọng số này, khi phân tách câu sẽ quyết định một cụm các tiếng có phải là một từ hay không.
- Hướng tiếp cận này tỏ ra linh hoạt hơn so với hướng tiếp cận dựa vào từ điển, tuy nhiên nó lại phụ thuộc vào dữ liệu học ban đầu và cần có thời gian để tích lũy.
- Hiện nay đã có 1 hướng tiếp cận mới là dựa vào thống kê Internet, phương pháp này sử dụng các search engine hiện có như google, bing,… Dựa vào kết quả tìm kiếm, thuật toán sẽ đánh giá mức độ liên kết giữa các từ (Mutual information - MI) và sử dụng nó để quyết định có phải là từ hay không. Đây là hướng tiếp cận hứa hẹn có nhiều triển vọng.

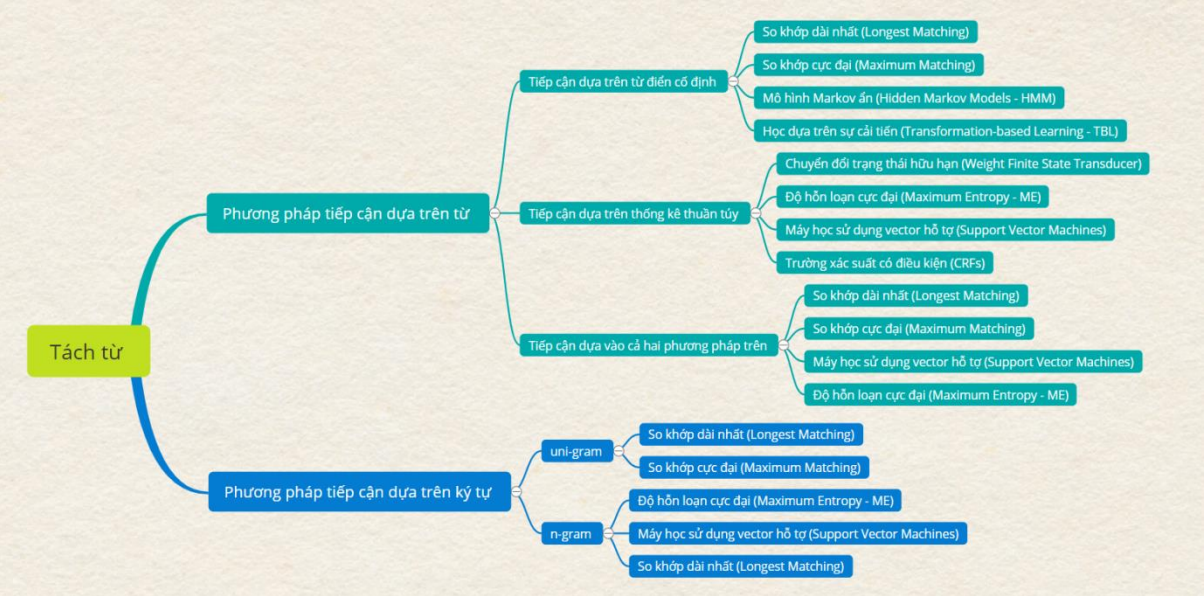
Tiếp cận dựa vào cả hai phương pháp trên:

- Hướng tiếp cận này kết hợp sử dụng cả từ điển và thống kê tận dụng ưu điểm của cả 2 hướng tiếp cận trên. Tuy có ưu điểm về độ chính xác nhưng nó lại gặp phải các bấn đề phức tạp khác, đòi hỏi nhiều hơn về thời gian cũng như bộ nhớ.

Phương pháp tiếp cận dựa trên ký tự

- Phương pháp này có thể được chia làm 2 nhóm nhỏ:uni-gram và n-gram. Hướng tiếp cận dựa trên một ký tự (uni-gram) chia văn bản ra các ký tự đơn lẻ để thực hiện việc tách từ.

  Hướng tiếp cận dựa trên nhiều ký tự (n-gram) chia văn bản thành nhiều chuỗi, mỗi chuỗi gồm hai, ba ký tự trở lên. So với hướng tiếp cận dựa trên một ký tự, hướng tiếp cận này cho nhiều kết quả ổn định hơn (cụ thể là trong tiếng Hoa).
- Khái niệm ký tự ở đây được tương đương với tiếng trong tiếng Việt. Phương pháp này chia 1 câu ra thành các tiếng phân cách nhau bởi các dấu cách hay các dấu câu, sau đó dựa vào các giải thuật như quy hoạch động, giải thuật di truyền hay thống kê để tìm ra cách phân chia mà có tổng xác suất các phân đoạn là tối ưu nhất.
- Phương pháp này tỏ ra đơn giản, linh hoạt hơn do không phải dựa vào một bộ từ điển cố định. Cách tiếp cận này có tính khả quan, hứa hẹn nhiều triển vọng hơn so với phương pháp tiếp cận dựa trên từ điển.



Các phương pháp tách từ

3. Phương pháp longest matching

Định nghĩa

Trong các hướng tiếp cận tách từ thì so khớp dài nhất (longest matching) thuộc vào cách tiếp cận dựa vào từ điển cố định.

Phương pháp này sẽ duyệt câu từ trái qua phải hoặc từ phải qua trái, lần lượt duyệt chuỗi các tiếng kiểm tra xem nó có phải là từ hay không. Chuỗi dài nhất được xác định là từ sẽ được chọn ra, tiếp tục làm như thế với chuỗi còn lại của câu cho đến khi hết câu.

Đây là dạng đơn giản của phương pháp Maximum matching, dạng phức tạp sẽ là tìm chuỗi phân đoạn dài nhất của 3 từ liền nhau.

Cách thực hiện

Các câu đầu vào ban đầu sẽ được phân tách thành một chuỗi các tiếng, chẳng hạn với câu A ta phân tích được chuỗi các tiếng là: A1 A2 A3 …

Sau đó, một vòng lặp sẽ được khởi tạo để so khớp các từ trong câu với từ điển sẵn có. Mục tiêu là cho việc xác định chuỗi dài nhất có thể có trong câu. Ví dụ:

+ Xác định A1 có phải là từ hay không

+ Xác định A1A2 có phải là từ hay không

+ Xác định A1A2A3 có phải là từ hay không

+ …

⇨ Chuỗi dài nhất được xác định sẽ là từ được chọn.

Tiếp tục xét chuỗi các tiếng còn lại cho đến khi xét hết chuỗi thì dừng lại.

Yêu cầu

Bộ từ điển bao gồm các từ cần tách

Chuỗi đầu vào đã tách các dấu câu và âm tiết.

Tư tưởng

Longest matching là thuật toán tham lam, phương pháp này sẽ tách các chuỗi dài nhất tìm được để tối ưu hóa bài toán nhưng có thể sẽ không chính xác với vài trường hợp nhất định.

Thuật toán có thể đi từ trái sang phải hoặc đi từ phải sang trái, lấy các từ dài nhất có thể, dừng lại khi duyệt hết. Việc bắt đầu từ các vị trí khác nhau có thể sẽ có kết quả khác nhau.

Độ phức tạp: O(n,V). Trong đó:

+ n: Số âm tiết trong chuỗi.

+ V: Số từ trong từ điển.

Xây dựng thuật toán

Ta có thể xây dựng thuật toán longest matching như sau:

Begin

1. Cho chuỗi đầu vào [w_0 w_1 … w_n-1]
2. Words = []
3. s = 0
4. e = n
5. Khi [w_s … w_e] chưa là một từ e = e -1
6. Words =Words + [w_s … w_e]
7. s = e + 1
8. Nếu e < n: Quay lại bước (4)
9. Lấy ra chuỗi đã tách từ Words

End

Ví dụ minh họa

Với chuỗi đầu vào là: "môn học xử lý ngôn ngữ tự nhiên", ta sẽ có được quá trình tách từ sử dụng phương pháp longest matching như sau:

- Chuỗi đầu vào: "môn học xử lý ngôn ngữ tự nhiên"
- Xây dựng từ điển: {"môn học": 0, "môn":1, "học": 2, "xử lý": 3, "ngôn ngữ": 4, "tự nhiên": 5}

Xử lý:

- B1: Duyệt chuỗi từ phải sang trái.

  Đối chiếu từ điển ta tìm được từ dài nhất đầu tiên là: môn học
  - Tách từ đó ra ta có từ đầu tiên là: môn_học
  - Chuỗi sau khi tách được: môn_học
  - Chuỗi còn lại là: xử lý ngôn ngữ tự nhiên
- B2: Thực hiện duyệt chuỗi còn lại như B1.

  Ta có từ dài nhất là: xử lý.

- Tách từ đó ra khỏi chuỗi ta được từ: xử_lý
- Chuỗi sau khi tách được là: môn_học xử_lý
- Chuỗi còn lại là: ngôn ngữ tự nhiên

- B3: Thực hiện tương tự B1 trên chuỗi còn lại.

  Ta có từ dài nhất trong chuỗi là: ngôn ngữ

  - Tách từ đó ra khỏi chuỗi ta được từ: ngôn_ngữ
  - Chuỗi sau khi tách được là: môn_học xử_lý ngôn_ngữ
  - Chuỗi còn lại là: tự nhiên
- B4: Tiếp tục thực hiện trên chuỗi còn lại.

  Chuỗi còn lại cũng là từ dài nhất: tự nhiên

  - Tách nó ra khỏi chuỗi còn lại ta được: tự_nhiên
  - Chuỗi sau khi tách được là: môn_học xử_lý ngôn_ngữ tự_nhiên
  - Chuỗi còn lại là chuỗi rỗng => kết thúc và xuất ra kết quả tách được.

Kết quả: môn_học xử_lý ngôn_ngữ tự_nhiên

Lần tách	Chuỗi còn lại	Từ tách được	Chuỗi có được sau khi tách
1	môn học xử lý ngôn ngữ tự nhiên	môn học	môn_học
2	xử lý ngôn ngữ tự nhiên	xử lý	môn_học xử_lý
3	ngôn ngữ tự nhiên	ngôn ngữ	môn_học xử_lý ngôn_ngữ
4	tự nhiên	tự nhiên	môn_học xử_lý ngôn_ngữ tự_nhiên

Ví dụ về phần tách từ theo phương pháp Longest matching

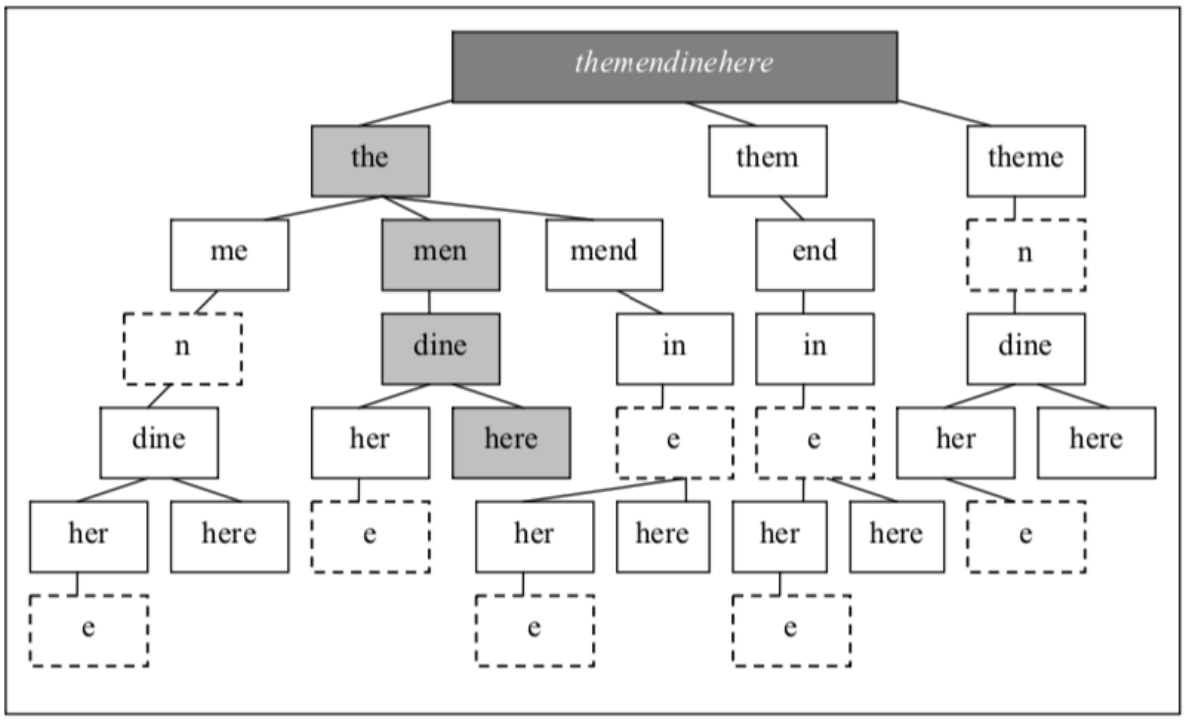
Ưu điểm và nhược điểm

Ưu điểm:

- Cách tách từ đơn giản, nhanh chóng và dễ thực hiện, chỉ cần dựa vào từ điển.
- Cài đặt đơn giản.
- Độ phức tạp tính toán hợp lý.
- Không yêu cầu dữ liệu huấn luyện.
- Độ chính xác cao. Nếu bộ ngữ liệu đơn giản và hầu như không có vấn đề nhập nhằng thì độ chính xác có thể lên tới 100%.

Nhược điểm:

- Phương pháp này phụ thuộc hoàn toàn vào độ chính xác và đầy đủ của từ điển → xây dựng được bộ từ điển đầy đủ là vấn đề cốt lõi. Mục tiêu này khá khó khăn do kho từ vựng dân gian là vô cùng lớn
- Chưa giải quyết được vấn đề nhập nhằng.



Tách từ sử dụng longest matching

II. Gán nhãn từ loại

1. Markov chain

Xích Markov (Markov chain) hay chuỗi Markov là một mô hình ngẫu nhiên mô tả một chuỗi các sự kiện có thể xảy ra, trong đó xác suất của mỗi sự kiện chỉ phụ thuộc vào trạng thái của sự kiện trước đó. Xích Markov có thể là một chuỗi vô hạn đếm được, trong đó các sự kiện diễn ra rời rạc, tạo thành xích Markov rời rạc. Nếu các sự kiện diễn ra liên tục thì quá trình đó gọi là xích Markov liên tục.

Xích Markov có nhiều ứng dụng trong các mô hình thống kê quy trình xảy ra ở thế giới thực, chẳng hạn như nghiên cứu hệ thống kiểm soát hành trình trên xe cơ giới, hàng đợi, tỷ giá thị trường chứng khoán và động thái quần thể động vật.

Đặc điểm của một xích Markov được biểu diễn bởi xác suất có điều kiện $P(X_{n+1} | X_n)$. Cụ thể là xác suất chuyển sang trạng thái tiếp theo chỉ phụ thuộc vào trạng thái hiện tại chứ không phụ thuộc vào các trạng thái trước đó. Đây gọi là ***thuộc tính Markov***.

Để làm rõ hơn về xích Markov ta có ví dụ đơn giản về thời tiết như sau:

- Nếu thời tiết hôm nay là trời nắng. Xác suất ngày mai trời nắng là 0.5, trời mưa là 0.3, trời mây là 0.2
- Nếu thời tiết hôm nay là trời mưa. Xác suất ngày mai trời nắng là 0.3, trời mưa là 0.7, trời mây là 0
- Nếu thời tiết hôm nay là trời mây. Xác suất ngày mai trời nắng là 0.5, trời mưa là 0.5, trời mây là 0

Xác suất điều kiện của thời tiết (được mô hình hóa là nắng, mưa hoặc mây), dựa trên thời tiết của ngày hôm trước, có thể được biểu diễn bằng ma trận chuyển trạng thái (transition matrix) như sau:

	Nắng	**Mưa**	**Mây**
Nắng	0.5	0.3	0.2
Mưa	0.3	0.7	0

Mây	0.5	0.5	0

Lưu ý rằng tổng của các hàng trong ma trận đều bằng 1: do đó là ***ma trận ngẫu nhiên.***

Giả sử trong ba ngày liên tục mà chuỗi thời tiết diễn ra như thế này:

- Nắng → Mây → Nắng

Thì liệu có thể tìm được xác suất của thời tiết ở ngày 4

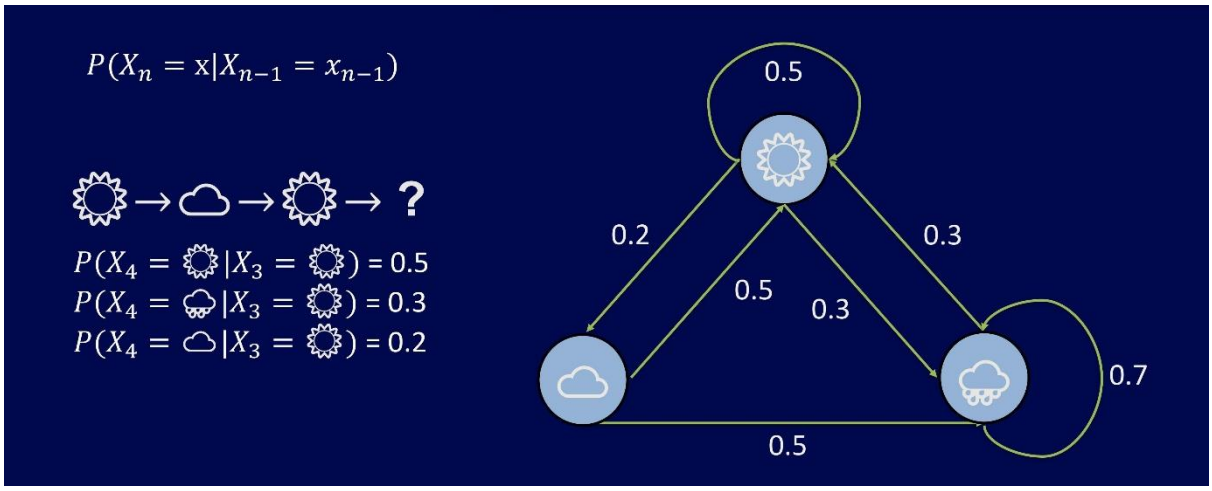
- Nắng → Mây → Nắng → ?

Thông thường thì xác suất thời tiết ngày 4 sẽ được tính như sau: $P(x_4 \mid x_1,x_2,x_3)$.

Nhưng do thuộc tính của xích Markov là xác suất xảy ra trạng thái hiện tại chỉ phụ thuộc vào trạng thái trước đó: $P(X_{n+1} \mid X_n)$. Do đó xác suất thời tiết ở ngày 4 chỉ phụ thuộc vào ngày 3. Xác suất thời tiết ngày 4 có thể viết lại như sau: $P(x_4 \mid x_3)$.

Vì vậy, các xác suất xảy ra các thời tiết nắng, mưa hoặc mây ở ngày 4 với thời tiết ngày 3 là nắng sẽ có kết quả lần lượt là:

- $P(x_4 = \text{nắng} \mid x_3 = \text{nắng}) = 0.5$
- $P(x_4 = \text{mưa} \mid x_3 = \text{nắng}) = 0.3$
- $P(x_4 = \text{mây} \mid x_3 = \text{nắng}) = 0.2$



Một ví dụ của Markov chain

Báo cáo này sẽ không đi chi tiết vào Markov chain và sẽ dừng lại tại đây do mô hình được xây trong báo cáo này không dựa trên Markov chain mà chỉ bắt nguồn từ khái niệm đó.

2. Mô hình Markov ẩn

Giới thiệu

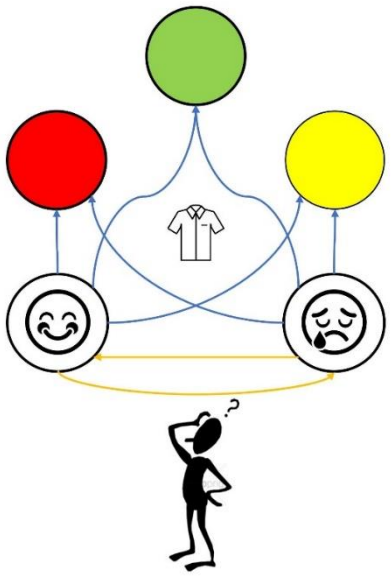
Mô hình Markov ẩn (Hidden Markov model) – có nguồn gốc từ Markov chain – là một mô hình thống kê trong đó hệ thống được mô hình hóa được xem như một xích Markov với các tham số ẩn. Ngoài ra, HMM còn có các tham số có thể quan sát được. Các tham số quan sát được sẽ có một sự liên quan nhất định với các tham số ẩn. Do các tham số ẩn không thể đánh giá trực tiếp được nên mục tiêu của mô hình Markov ẩn là sẽ tìm hiểu các tham số ẩn một cách gián tiếp thông qua các tham số quan sát được. Để hiểu được khái niệm này ta sẽ xét một ví dụ như sau:

- Giả sử có một bạn X. Mỗi ngày bạn đó sẽ mặc một trong ba áo có màu là đỏ, vàng, xanh. Màu áo X sẽ chọn phụ thuộc vào tâm trạng hôm đó: vui hoặc buồn và tâm trạng ngày hôm trước sẽ phụ thuộc vào ngày hôm nay.

Dựa vào mô hình Markov ẩn, ta có thể trình bày như sau:

- Trạng thái quan sát được là màu áo của X
- Trạng thái ẩn là tâm trạng của X

Vậy yêu cầu đặt ra là hãy dự đoán chuỗi tâm trạng trong các ngày có khả năng nhất từ màu áo mà X chọn trong ngày đó



Một ví dụ về mô hình Markov ẩn

Thành phần

Một mô hình Markov ẩn sẽ chứa các thành phần:

- N: Số trạng thái ẩn.
- M: Số trạng thái quan sát được.
- T: Độ dài của chuỗi trạng thái quan sát được.
- i_t: Trạng thái ẩn tại thời điểm t.
- $V = \{v_1, v_2 \dots .v_m\}$: Chuỗi rời rạc các trạng thái quan sát được.
- $s = \{s_i\}$: Xác suất xảy ra trạng thái ẩn i lúc bắt đầu
- $A = \{a_{ij}\}$: Ma trận chuyển trạng thái A (transition matrix), trong đó $a_{ij} = P(i_{t+1} = j \mid i_t = i)$ là xác suất chuyển sang trạng thái ẩn j tại thời điểm t + 1 từ trạng thái i tại thời điểm t.
- $B = \{b_j(k)\}$: Ma trận thể hiện, trong đó $b_j(k) = P(v_k \text{ tại } t \mid i_t = j)$ là xác suất xảy ra của trạng thái quan sát được v_k từ trạng thái ẩn j
- O_t: Trạng thái quan sát được tại thời điểm t.
- $\lambda = (A, B, s)$ là tập biểu diện một mô hình Markov ẩn.

Thuật toán

Giả sử ta có một tập hữu hạn các trạng thái quan sát được và một tập hữu hạn các trạng thái ẩn. Khi đó tập S sẽ là tập hợp tất cả các cặp trạng thái ẩn và trạng thái quan sát được. S = (x_1, x_2, x_3 …, x_n, y_1, y_2, y_3, …, y_n), sao cho n > 0, x_i là các trạng thái thuộc tập quan sát được và y_i là các trạng thái ẩn.

Khi đó:

- Với mọi giá trị (x_1, x_2, x_3 …, x_n, y_1, y_2, y_3, …, y_n) ϵ S, p(x_1, x_2, x_3 …, x_n, y_1, y_2, y_3, …, y_n) $\geq$ 0.
- $\sum p(x_1, x_2, x_3 \dots, x_n, y_1, y_2, y_3 \dots, y_n) = 1$

Vậy, lúc này ta cần tính xác suất cao nhất của chuỗi trạng thái ẩn có thể xảy ra từ chuỗi các trạng thái quan sát được cho trước, công thức tính là:

- Argmax p(x_1, x_2, x_3 …, x_n, y_1, y_2, y_3, …, y_n)

Theo toán học, xác suất đó có thể được viết thành thế này:

- Argmax p(x_1, x_2, x_3 …, x_n, y_1, y_2, y_3, …, y_n) = p(y_1)p(y_2| y_1)p(y_3| y_2, y_1) …p(y_n| y_{n-1}, …, y_2, y_1)p(x_1| y_{n-1}, …, y_2, y_1)p(x_2| x_1, y_{n-1}, …, y_2, y_1) p(x_3| x_2, x_1, y_{n-1}, …, y_2, y_1) …p(x_n| x_{n-1}, …, x_2, x_1, y_{n-1}, …, y_2, y_1)

Nhưng do mô hình Markov ẩn có thừa hưởng thuộc tính Markov từ Markov chain nên ta có thể viết các xác suất trong công thức lại như thế này:

$$\begin{cases} p(x_n|x_{n-1}, \dots, x_2, x_1, y_n, y_{n-1}, \dots, y_2, y_1) \\ \dots \\ p(x_3|\, x_2, x_1, y_n, y_{n-1}, \dots, y_2, y_1) \\ p(x_2|\, x_1, y_n, y_{n-1}, \dots, y_2, y_1) \\ p(x_1|y_n, y_{n-1}, \dots, y_2, y_1) \\ p(y_n|\, y_{n-1}, \dots, y_2, y_1) \\ \dots \\ p(y_3|y_2, y_1) \\ p(y_2|\, y_1) \\ p(y_1) \end{cases} \Rightarrow \begin{cases} p(x_n|\, y_n) \\ \dots \\ p(x_3|\, y_3) \\ p(x_2|\, y_2) \\ p(x_1|\, y_1) \\ p(y_n|\, y_{n-1}) \\ \dots \\ p(y_3|\, y_2) \\ p(y_2|\, y_1) \\ p(y_1|s) \end{cases}$$

⇨ Argmax $p(x_1, x_2, x_3 \ldots, x_n, y_1, y_2, y_3, \ldots, y_n) = p(y_1|\ s)p(y_2|\ y_1)\ p(y_3|\ y_2)$ $\ldots p(y_n|\ y_{n-1})p(x_1|\ y_1)\ p(x_2|\ y_2)\ p(x_3|\ y_3)\ldots\ p(x_n|\ y_n)$

** s là trạng thái bắt đầu*

Trong đó:

- $p(y_i|\ y_{i+1})$ là các xác suất trong ma trận chuyển trạng thái (transition matrix)
- $p(x_i|\ y_i)$ là các xác suất trong ma trận thể hiện(emission matrix)

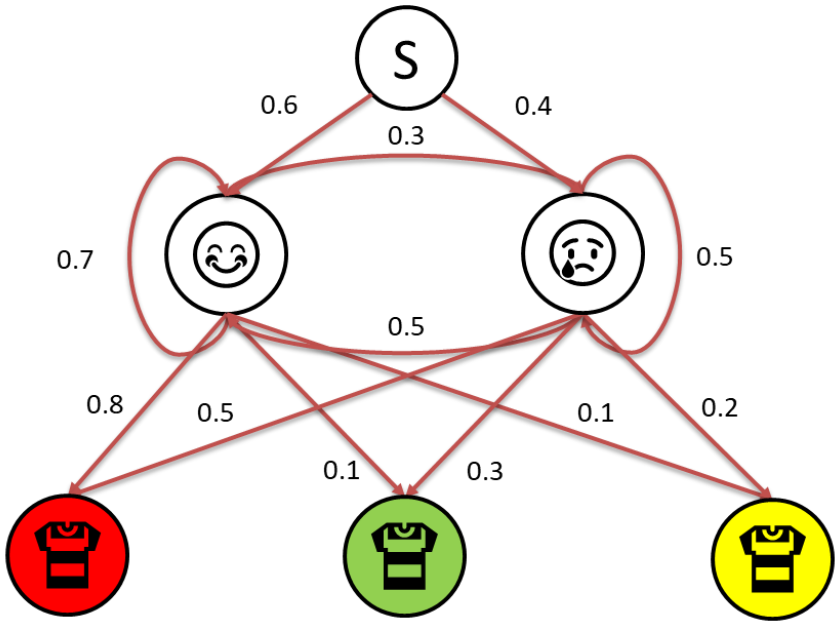
Vậy, ta có thể trình bày thuật toán của mô hình Markov ẩn một cách tổng quát như sau:

$$\text{argmax}\prod_{i=1}^{n} p(y_i|y_{i+1}) \prod_{i=1}^{n} p(x_i|y_i)$$

Ví dụ làm rõ

Để làm rõ về quá trình tính toán của mô hình Markov ẩn, ta hãy trở lại ví dụ về màu áo bạn X sẽ mặc dựa trên tâm trạng hôm đó nhưng lần này sẽ có các xác suất tương ứng được thêm vào bài toán.

Ta sẽ có một lược đồ xích Markov như sau:



Tóm tắt lược đồ:

- Tại lúc đầu, xác suất bắt đầu một ngày vui của X là 0.6, buồn là 0.4
- Nếu ngày hôm đó X vui thì xác suất ngày mai vui là 0.7, buồn là 0.3
- Nếu ngày hôm đó X buồn thì xác suất ngày mai vui là 0.5, buồn là 0.5
- Xác suất trong ngày vui X mặc áo đỏ là 0.8, xanh là 0.1, vàng là 0.1
- Xác suất trong ngày buồn X mặc áo đỏ là 0.5, xanh là 0.3, vàng là 0.2

Ma trận chuyển trạng thái tương ứng:

	Vui	**Buồn**
Vui	0.7	0.3
Buồn	0.5	0.5

Ma trận thể hiện tương ứng:

	Đỏ	**Xanh**	**Vàng**
Vui	0.8	0.1	0.1
Buồn	0.5	0.3	0.2

Câu hỏi đặt ra: Giả sử trong ba ngày liên tục, X mặc ba màu tương ứng theo thứ tự từng ngày như sau: Xanh → Vàng → Đỏ thì chuỗi tâm trạng nào cho từng ngày có xác suất lớn nhất mà X có thể có.

Vậy ta cần tính: argmax p(xanh, vàng, đỏ, m_1, m_2, m_3) với m_1, m_2, m_3 là tâm trạng tương ứng của X với từng ngày 1, 2, 3

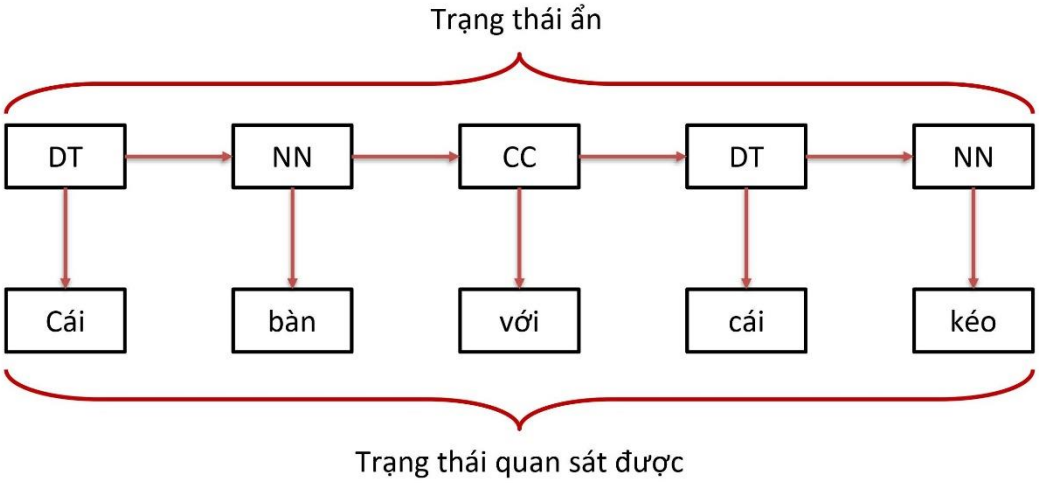
Các kết quả:

- p(xanh, vàng, đỏ, vui, vui, vui) =
  p(đỏ| vui)p(vàng| vui)p(xanh| vui)p(vui| vui)p(vui| vui)p(vui| s) =
  $0.8 \times 0.1 \times 0.1 \times 0.7 \times 0.7 \times 0.6 = 2.352 \times 10^{-3}$
- p(xanh, vàng, đỏ, vui, vui, buồn) =
  p(đỏ| buồn)p(vàng| vui)p(xanh| vui)p(buồn| vui)p(vui| vui)p(vui| s) =
  $0.5 \times 0.1 \times 0.1 \times 0.3 \times 0.7 \times 0.6 = 0.63 \times 10^{-3}$
- p(xanh, vàng, đỏ, vui, buồn, buồn) =
  p(đỏ| buồn)p(vàng| buồn)p(xanh| vui)p(buồn| buồn)p(buồn|vui)p(vui| s) =
  $0.5 \times 0.2 \times 0.1 \times 0.5 \times 0.3 \times 0.6 = 0.9 \times 10^{-3}$
- p(xanh, vàng, đỏ, buồn, buồn, buồn) =
  p(đỏ|buồn)p(vàng|buồn)p(xanh|buồn)p(buồn|buồn)p(buồn|buồn)p(buồn|s) =
  $0.5 \times 0.2 \times 0.3 \times 0.5 \times 0.5 \times 0.4 = 3 \times 10^{-3}$
- p(xanh, vàng, đỏ, buồn, buồn, vui) =
  p(đỏ|vui)p(vàng|buồn)p(xanh|buồn)p(vui|buồn)p(buồn|buồn)p(buồn|s) =
  $0.8 \times 0.2 \times 0.3 \times 0.5 \times 0.5 \times 0.4 = 4.8 \times 10^{-3}$ → Lớn nhất
- p(xanh, vàng, đỏ, buồn, vui, vui) =
  p(đỏ|vui)p(vàng|vui)p(xanh|buồn)p(vui|vui)p(vui|buồn)p(buồn|s) =
  $0.8 \times 0.1 \times 0.3 \times 0.7 \times 0.5 \times 0.4 = 3.36 \times 10^{-3}$
- p(xanh, vàng, đỏ, buồn, vui, buồn) =
  p(đỏ|buồn)p(vàng|vui)p(xanh|buồn)p(buồn|vui)p(vui|buồn)p(buồn|s) =
  $0.5 \times 0.1 \times 0.3 \times 0.3 \times 0.5 \times 0.4 = 0.9 \times 10^{-3}$
- p(xanh, vàng, đỏ, vui, buồn, vui) =
  p(đỏ|vui)p(vàng|buồn)p(xanh|vui)p(vui|buồn)p(buồn|vui)p(vui|s) =
  $0.8 \times 0.2 \times 0.1 \times 0.5 \times 0.3 \times 0.6 = 1.44 \times 10^{-3}$

Trong các xác suất tính được, ta có thể thấy xác suất của p(xanh, vàng, đỏ, buồn, buồn, vui) là lớn nhất. Do đó chuỗi tâm trạng có khả năng xảy ra nhất là {buồn, buồn, vui}.

Mô hình Markov ẩn trong gán nhãn

Trong gán nhãn, mô hình Markov ẩn là một trong những phương pháp đơn giản và linh hoạt nhất. Nếu cho trước một câu cần gán nhãn thì các từ trong câu sẽ được xem như trạng thái quan sát được còn các nhãn tương ứng là các trạng thái ẩn.



HMM trong POS tagging

Vấn đề

Nhắc lại về HMM:

- Thuật toán: Max $P(p_1, \dots, p_n, w_1, \dots, w_n)$
- Trong đó: +) p_i là nhãn i tương ứng.

  +) w_i là từ i tương ứng.

Nếu xét lại ví dụ đã cho về tâm trạng của X ở trên ta thấy với 2 trạng thái ẩn và 3 trạng thái quan sát được, ta có $2^3 = 8$ khả năng. Nhìn có vẻ tuy không nhiều nhưng nếu tăng số trạng thái ẩn lên thành 3 và trạng thái quan sát được lên thành 4 thì số khả năng là $3^4 = 81$ → Số lượng sẽ tăng theo cấp số nhân

Trong gán nhãn, chỉ cần một câu đơn giản gồm 5 nhãn và 10 từ thì số khả năng cần phải tính toán là $5^{10} \approx 10$ triệu *Không khả quan

Sự tăng theo cấp số nhân sẽ là vấn đề đối với bất kỳ bộ ngữ liệu lớn nào. Do đó, cách tiếp cận thông thường sẽ không hiệu quả vì sẽ mất quá nhiều thời gian để thực hiện.

Thuật toán Viterbi

Để giải quyết vấn đề nêu trên, thay vì cách tiếp cận thông thường, chuỗi nhãn có khả năng xảy ra cao nhất có thể được tìm ra một cách hiệu quả hơn bằng cách sử dụng một thuật toán lập trình động được gọi là thuật toán Viterbi.

Thuật toán Viterbi là một thuật toán tham lam do nó loại bỏ một cách có hệ thống những khả năng không thể xảy ra nếu trong quá trình tính toán chúng có xác suất xuất hiện nhỏ hơn các khả năng khác. Vì tất cả các khả năng không thể xảy ra đã bị loại bỏ, thuật toán chỉ cần theo dõi chuỗi trạng thái có khả năng xảy ra nhất. Khi trạng thái hiện tại được nhận, xác suất của chuỗi trạng thái tại thời điểm hiện tại được tính bằng cách thêm các xác suất của chuỗi trạng thái trước có khả năng xảy ra vào. Sau đó, các chuỗi dẫn vào mỗi trạng thái tại thời điểm hiện tại được so sánh và chuỗi nào có xác suất cao nhất sẽ là kết quả cuối cùng.

Để tường minh hơn, chúng tôi xin mô tả quá trình tìm ra nhãn cho một câu ví dụ tiếng Việt cụ thể, chẳng hạn câu: *Cái bàn với cái kéo*

Giả sử sau khi huấn luyện bộ ngữ liệu ta có ma trận chuyển trạng thái và ma trận thể hiện sau:

- Ma trận chuyển trạng thái:

	DT	NN	CC	VB
< S >	0.7	0.2	0.1	0
DT	0	0.2	0.3	0.5
NN	0	0.25	0.5	0.25

CC	0.7	0.1	0	0.2
VB	0.25	0.5	0.25	0

- Ma trận thể hiện

	cái	**bàn**	**với**	**kéo**
DT	0.3	0	0	0
NN	0	0.3	0	0.5
CC	0	0	0.3	0
VB	0	0.2	0.25	0.1

Trình tự tính toán chuỗi nhãn sẽ diễn ra như sau:

- Nhãn từ đầu tiên:

  Từ "cái" có một nhãn duy nhất là DT và là từ bắt đầu, xác suất:

  - chuỗi nhãn [DT] = p(cái|DT) p(DT|<S>) = 0.21

  ⇨ Chuỗi nhãn đầu tiên [DT]

- Nhãn từ thứ hai:

  Từ "bàn" có hai nhãn là NN hoặc VB và chuỗi nhãn trước là [DT], xác suất:

  - Chuỗi nhãn [DT, NN] = p([DT]) p(bàn|NN) p (NN|DT) = 0.0126
  - Chuỗi nhãn [DT, VB] = p([DT]) p(bàn|VB) p (VB|DT) = 0.021

  ⇨ Do chuỗi nhãn còn phụ thuộc vào khả năng của các nhãn sau nên chưa thể kết luận được nhãn từ hai

- Nhãn từ thứ ba:

  Từ "với" có hai nhãn là CC hoặc VB và chuỗi nhãn trước là [DT, NN] hoặc [DT, VB], xác suất:

  - Chuỗi nhãn [DT, NN, CC] = p([DT, NN]) p(với|CC) p(CC|NN) = 1.89×10^{-3}
  - Chuỗi nhãn [DT, NN, VB] = p([DT, NN]) p(với|VB) p(VB|NN) = 0.7875×10^{-3}
  - Chuỗi nhãn [DT, VB, CC] = p([DT, VB]) p(với|CC) p(CC|VB) = 1.575×10^{-3}
  - Chuỗi nhãn [DT, VB, VB] = p([DT, VB]) p(với|VB) p(VB|VB) = 0

  ⇨ Hai chuỗi nhãn thứ ba có khả năng là [DT, NN, CC] và [DT, NN, VB]. Ta có thể loại bỏ hai chuỗi [DT, VB, CC] và [DT, VB, VB] do đó là các chuỗi nhãn có xác suất thấp hơn chuỗi còn lại trong quá trình gán nhãn từ thứ ba. Đây là bước quan trọng trong thuật toán Viterbi do nó thể hiện rõ bản chất của thuật toán.

- Nhãn từ thứ tư:

  Từ "cái" có một nhãn duy nhất là DT và chuỗi nhãn trước là [DT, NN, CC] hoặc [DT, NN, VB], xác suất:

  - Chuỗi nhãn [DT, NN, CC, DT] = p([DT, NN, CC]) p(cái|DT) p(DT|CC) = 3.969×10^{-4}
  - Chuỗi nhãn [DT, NN, VB, DT] = p([DT, NN, VB]) p(cái|DT) p(DT|VB) = 5.90625×10^{-5}

  ⇨ Chuỗi nhãn thứ tư có khả năng nhất là [DT, NN, CC, DT]. Ta có thể loại bỏ chuỗi [DT, NN, VB, DT]

- Nhãn từ thứ 5:

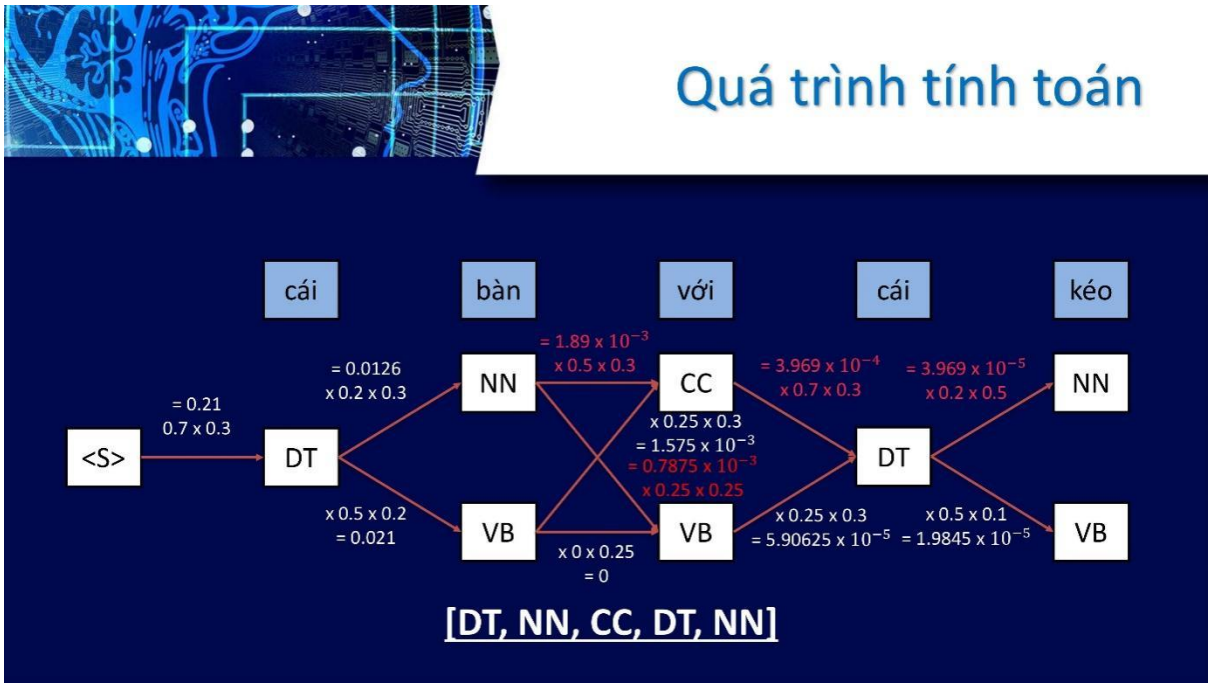
  Từ "kéo" có hai nhãn là NN hoặc VB và chuỗi nhãn trước là [DT, NN, CC, DT], xác suất:

- Chuỗi nhãn [DT, NN, CC, DT, NN] = p([DT, NN, CC, DT]) p(kéo|NN) p(NN|DT) = 3.969 x 10^{-5}
- Chuỗi nhãn [DT, NN, VB, DT, VB] = p([DT, NN, CC, DT]) p(kéo|VB) p(VB|DT) = 1.9845 x 10^{-5}

⇨ Do đây là nhãn từ cuối cùng nên ta có thể kết luận chuỗi nhãn thứ năm là [DT, NN, CC, DT, NN].

Vậy câu *cái bàn với cái ghế* có thể được gán nhãn như sau: cái/DT bàn/NN với/CC cái/DT ghế/NN

Bài toán có thể được mô hình hóa như sau:



Ví dụ về thuật toán Viterbi

**Trong ví dụ này, nhóm đã sử dụng bộ nhãn Penn Treebank cho việc gán nhãn câu tiếng Việt nên có thể gây ra sự nhầm lẫn do bộ Penn Treebank chuyên dùng cho tiếng Anh. Ngoài ra, có những từ cũng bị lược bỏ bớt nhãn như từ "cái". Trên thực tế, khi làm việc với bộ ngữ liệu lớn hơn qua các quá trình xử lý thì các xác suất trong ma trận thể hiện sẽ có tổng bằng 1 nhưng trong trường hợp này không như vậy. Các sai sót*

này là hoàn toàn có chủ đích với mục tiêu là có thể tường minh hóa cách mà thuật toán Viterbi vận hành. Mong người đọc sẽ bỏ qua.

Khi so với cách tính thông thường của mô hình Markov ẩn thì thuật toán Viterbi đã làm cho quá trình tính toán ngắn hơn rất nhiều do các khả năng không thể xảy ra đã bị loại bỏ

Nếu cho một câu có độ dài là L và số nhãn là P, thì các chuỗi nhãn có khả năng phải tính nếu sử dụng:

- Cách tính thông thường: P^L
- Thuật toán Viterbi: LP^2

Vậy ta có thể thấy thuật toán Viterbi đã làm giảm độ phức tạp đi rất nhiều.

Các thuật toán khác

Ngoài thuật toán Viterbi, còn có các thuật toán khác được sử dụng trong mô hình Markov ẩn phục vụ cho việc gán nhãn như sau:

- Thuật toán Baum–Welch
- Thuật toán Forward algorithm
- Thuật toán Forward

3. Smoothing

Vấn đề

Ta có một ví dụ như sau:

Cho một tập huấn luyện có các câu:

- bọn trẻ đang kéo nhau.
- cái bàn này vững lắm
- mọi người đang bàn với nhau
- cái bàn với cái kéo

Giả sử các câu trên trong tập huấn luyện được gán nhãn thế này:

- bọn/N trẻ/N đang/R kéo/V nhau/N
- cái/N bàn/N này/P vững/A
- mọi/D người/N đang/R bàn/V với/C nhau/N
- cái/N bàn/N với/C cái/N kéo/N

**Bộ nhãn được sử dụng là VLSP*

Từ bộ ngữ liệu trên thì ta có thống kê tần số xuất hiện các khả năng sau:

- Tần số chuyển từ nhãn này sang nhãn khác:

	N	**R**	**V**	**P**	**A**	**D**	**C**	**Tổng**
< S >	3	0	0	0	0	1	0	**4**
N	4	2	0	1	0	0	1	**8**
R	0	0	2	0	0	0	0	**2**
V	1	0	0	0	0	0	1	**2**
P	0	0	0	0	1	0	0	**1**
A	0	0	0	0	0	0	0	**0**
D	1	0	0	0	0	0	0	**1**
C	2	0	0	0	0	0	0	**2**

- Tần số từ gán với nhãn tương ứng:

	bọn	trẻ	đang	kéo	nhau	cái	bàn	này	vững	mọi	người	với	Tổng
N	1	1	0	1	2	3	2	0	0	0	1	0	**11**
R	0	0	2	0	0	0	0	0	0	0	0	0	**2**
V	0	0	0	1	0	0	1	0	0	0	0	0	**2**
P	0	0	0	0	0	0	0	1	0	0	0	0	**1**
A	0	0	0	0	0	0	0	0	1	0	0	0	**1**
D	0	0	0	0	0	0	0	0	0	1	0	0	**1**
C	0	0	0	0	0	0	0	0	0	0	0	2	**2**

Từ thống kê tần số, các xác suất có trong ma trận tương ứng có thể được tính bằng cách chia tần số cho tổng tương ứng.

Ta có ma trận chuyển trạng thái và ma trận thể hiện sau:

- Ma trận chuyển trạng thái:

	N	R	V	P	A	D	C
< S >	3/4	0	0	0	0	1/4	0
N	4/8	2/8	0	1/8	0	0	1//8
R	0	0	2/2	0	0	0	0
V	1/2	0	0	0	0	0	1/2
P	0	0	0	0	1/1	0	0
A	undef	undef	undef	undef	undef	undef	undef
D	1/1	0	0	0	0	0	0
C	2/2	0	0	0	0	0	0

- Ma trận thể hiện:

	bọn	trẻ	đang	kéo	nhau	cái	bàn	này	vững	mọi	người	với
N	1/11	1/11	0	1/11	2/11	3/11	2/11	0	0	0	1/11	0
R	0	0	2/2	0	0	0	0	0	0	0	0	0
V	0	0	0	1/2	0	0	1/2	0	0	0	0	0
P	0	0	0	0	0	0	0	1/1	0	0	0	0
A	0	0	0	0	0	0	0	0	1/1	0	0	0
D	0	0	0	0	0	0	0	0	0	1/1	0	0
C	0	0	0	0	0	0	0	0	0	0	0	2/2

Ta có thể thấy các vấn đề như sau:

- Trong ma trận thể hiện, các nhãn chuyển từ nhãn A đều là undefined (không xác định) do trong phần thống kê không có nhãn nào chuyển từ nhãn A. Điều này sẽ gây nhiễu bộ dữ liệu.
- Trong cả hai ma trận, có các xác suất bằng 1 như p(V|R), p(này|P). Có nghĩa là các khả năng này sẽ luôn xảy ra. Nhưng trong thực tế, khi bộ ngữ liệu đầu vào lớn hơn thì sẽ có nhiều khả năng khác chứ không phải duy nhất một khả năng.
- Phần lớn các xác suất đều bằng 0. Khác với vấn đề ở trên thì các khả năng này sẽ không bao giờ xảy ra. Tuy nhiên vẫn có thể xuất hiện nếu bộ ngữ liệu đủ lớn.

Để giải quyết các vấn đề gây nhiễu trên và làm cân bằng lại bộ dữ liệu sao cho hợp lý, ta sẽ xét đến phương pháp smoothing.

Khái quát

Smoothing (làm mịn) là một kỹ thuật toán học loại bỏ sự thay đổi dữ liệu dư thừa trong khi duy trì đánh giá đúng về bộ dữ liệu. Điều này cho phép các mẫu và xu hướng quan trọng trở nên nổi bật hơn.

Trong HMM, smoothing được sử dụng để cải thiện các ước tính xác suất. Smoothing sẽ phân phối các xác suất sao cho tất cả các chuỗi nhãn đều có thể xảy ra với khả năng nhất định. Điều này liên quan đến việc phân phối lại bộ dữ liệu sao cho không có xác suất nào chiếm ưu thế cũng như không có xác suất nào bằng 0.

"Bất cứ khi nào sự thưa thớt dữ liệu là một vấn đề, smoothing có thể giúp tăng hiệu suất. Sự thưa thớt dữ liệu hầu như luôn là một vấn đề trong mô hình thống kê. Trong các trường hợp cực đoan như khi có quá nhiều dữ liệu huấn luyện mà tất cả các tham số có thể được huấn luyện chính xác mà không cần smoothing, ta luôn luôn có thể mở rộng mô hình, chẳng hạn bằng cách chuyển sang mô hình n-gram bậc cao hơn, để đạt được hiệu suất cao hơn. Với nhiều thông số hơn, việc thưa thớt dữ liệu lại trở thành một vấn đề nhưng với việc smoothing thích hợp, các mô hình sẽ thường chính xác hơn các mô hình ban đầu. Vì vậy, dù bộ dữ liệu có lớn đến thế nào, smoothing hầu như luôn có thể giúp tăng hiệu suất với nỗ lực tương đối nhỏ."

Chen & Goodman (1998)

Các phương pháp smoothing

Laplace smoothing:

- Còn được biết đến là phương pháp cộng 1.
- Để tránh bất kỳ xác suất nào bằng 0 đối với các sự kiện không bao giờ xảy ra. Ta sẽ thực hiện như sau với phương pháp Laplace:

$$p(w_i|w_{i-1}) = \frac{1+c(w_{i-1}w_i)}{\sum_{w_i}[1+c(w_{i-1}w_i)]} = \frac{1+c(w_{i-1}w_i)}{|V|+\sum_{w_i} c(w_{i-1}w_i)}$$

- Khi áp dụng Laplace smoothing vào ví dụ trong phần vấn đề:

- Tần số chuyển từ nhãn này sang nhãn khác:

	N	R	V	P	A	D	C	Tổng
< S >	3 + 1	0 + 1	0 + 1	0 + 1	0 + 1	1 + 1	0 + 1	**4 + 7**
N	4 + 1	2 + 1	0 + 1	1 + 1	0 + 1	0 + 1	1 + 1	**8 + 7**
R	0 + 1	0 + 1	2 + 1	0 + 1	0 + 1	0 + 1	0 + 1	**2 + 7**
V	1 + 1	0 + 1	0 + 1	0 + 1	0 + 1	0 + 1	1 + 1	**2 + 7**
P	0 + 1	0 + 1	0 + 1	0 + 1	1 + 1	0 + 1	0 + 1	**1 + 7**
A	0 + 1	0 + 1	0 + 1	0 + 1	0 + 1	0 + 1	0 + 1	**0 + 7**
D	1 + 1	0 + 1	0 + 1	0 + 1	0 + 1	0 + 1	0 + 1	**1 + 7**
C	2 + 1	0 + 1	0 + 1	0 + 1	0 + 1	0 + 1	0 + 1	**2 + 7**

- Ma trận chuyển trạng thái:

	N	R	V	P	A	D	C
< S >	4/11	1/11	1/11	1/11	1/11	2/11	1/11
N	5/15	3/15	1/15	2/15	1/15	1/15	2/15
R	1/9	1/9	3/9	1/9	1/9	1/9	1/9
V	2/9	1/9	1/9	1/9	1/9	1/9	2/9
P	1/8	1/8	1/8	1/8	2/8	1/8	1/8
A	1/7	1/7	1/7	1/7	1/7	1/7	1/7
D	2/8	1/8	1/8	1/8	1/8	1/8	1/8
C	3/9	1/9	1/9	1/9	1/9	1/9	1/9

Quá trình này cũng áp dụng cho việc tính toán ma trận thể hiện.

Additive Smoothing:

- Phương pháp này khá giống Laplace smoothing. Thay vì thêm 1, một giá trị delta (δ) được cộng vào ước lượng hợp lý cực đại.
- Thông thường, $0 < \delta \leq 1$.

$$p_{add}(w_i|w_{i-n+1}^{i-1}) = \frac{\delta + c(w_{i-n+1}^{i})}{\delta|V| + \sum_{w_i} c(w_{i-n+1}^{i})}$$

Good – Turing Smoothing:

- Để tránh một cụm n-gram có nghĩa nhưng lại có tần suất xuất hiện nhỏ, người ta sử dụng phương pháp smoothing dựa trên ước lượng cùng tần suất, nghĩa là nhóm các cụm n-gram có số lần xuất hiện như nhau.

  - Với cụm n-gram chưa xuất hiện trước đó:

$$p_{unknown}\left(\frac{w_i}{w_{i-1}}\right) = \frac{N_1}{N}$$

  - Với cụm n-gram đã từng xuất hiện:

$$p_{\left(\frac{w_i}{w_{i-1}}\right)} = \frac{(c+1) * N_{c+1}}{N * N_c}$$

- Trong đó:
  - N_1: Số cụm n-gram xuất hiện 1 lần
  - N: Tổng số cụm n-gram
  - c: Số lần xuất hiện của cụm n-gram
  - N_{c+1}: Tổng số cụm n-gram xuất hiện c+1 lần

Các phương pháp smoothing khác:

- Jelinek – Mercer smoothing.
- Katz smoothing.
- Witten – Bell smoothing.
- Absolute discounting.
- Kneser – Ney smoothing

4. Tổng kết

HMM là phương pháp được nghiên cứu sâu, dễ hiểu, linh hoạt nên thích hợp cho giảng dạy, ứng dụng vào các bài toán thực tế ngay cả những vấn dề bên ngoài xử lý ngôn ngữ tự nhiên.

Tuy mô hình đơn giản do có "thuộc tính Markov", rằng trạng thái tiếp theo chỉ phụ thuộc vào trạng thái hiện tại. Tuy vậy, trong thực tế trạng thái tiếp theo có thể phụ thuộc vào trạng thái trước cả trạng thái hiện tại.

HMM không thể nắm bắt rõ ràng thời gian ở một trạng thái cụ thể do thuộc tính Markov. Tuy nhiên, mô hình Markov bán ẩn thì có thể làm được điều đó.

Chương 3. TẬP DỮ LIỆU

1. Cơ sở xây dựng

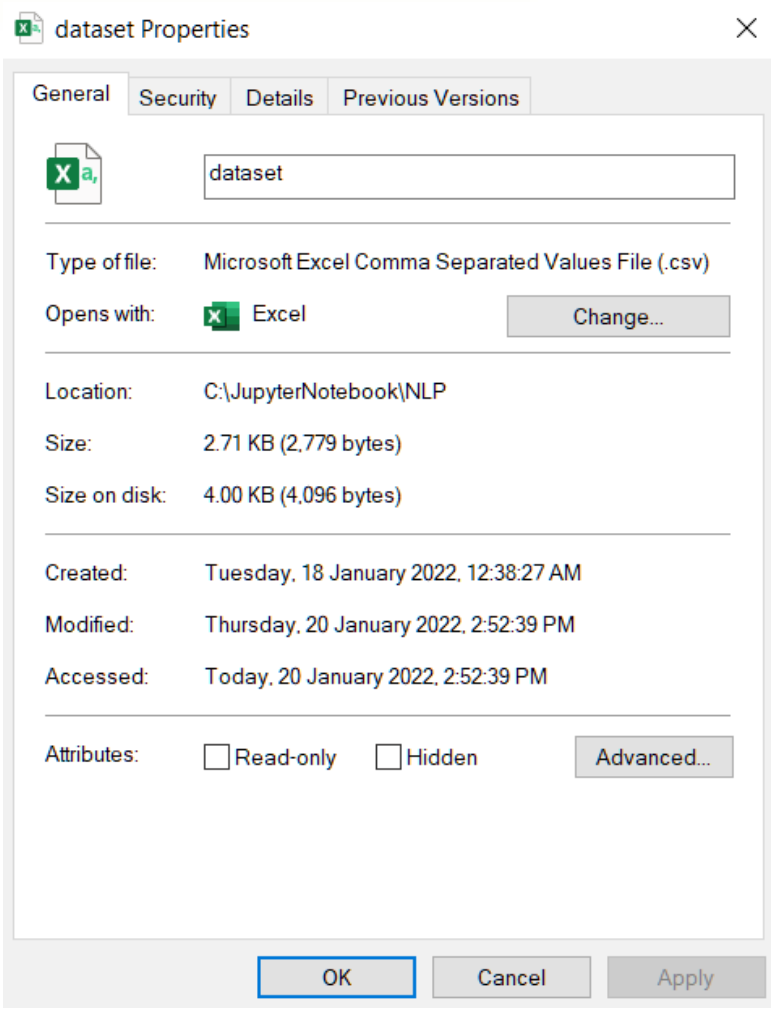
Tập dữ liệu sử dụng cho bài toán được nhóm xây dựng xuyên suốt quá trình học tập môn xử lý ngôn ngữ tự nhiên. Đó là các câu tiếng Việt rời rạc có liên quan với nhau về mặt từ ngữ. Nhóm đã thực hiện qua một số khâu tiền xử lý cơ bản trên bộ dữ liệu để có thể phục vụ nhu cầu bài toán.

2. Xây dựng tập dữ liệu cho bài toán

Trên cơ sở tập dữ liệu gốc, nhóm đã thực hiện qua một số khâu tiền xử lý như: chuẩn hóa các từ đặc biệt, kiểm tra các từ không trong bộ huấn luyện xuất hiện trong bộ thử nghiệm. Quy trình tiền xử lý này sẽ được trình bày rõ hơn ở các phần sau trong báo cáo này.

Quá trình xây dựng tập dữ liệu cho bài toán

Bộ dữ liệu gồm 40 câu tiếng Việt chưa được tách từ và 40 câu đã được tách từ thủ công lưu dưới dạng file.csv



Thông tin về tập dữ liệu

Sau đây là một số câu trong bộ dữ liệu:

1	đây là bài tập mới	đây là bài_tập mới
2	đây là một trong những bài tập khó	đây là một trong những bài_tập khó
3	nó đang làm bài tập	nó đang làm bài_tập
4	bài tập toán này dễ	bài_tập toán này dễ
5	cô ấy đang tập bài này	cô ấy đang tập bài này
6	chồng tập thật dày	chồng tập thật dày
7	hắn đang chồng đống bài tập	hắn đang chồng đống bài_tập

Ví dụ một số câu trong tập dữ liệu do nhóm xây dựng

Do đây chỉ là bộ dữ liệu nhỏ gồm 40 câu nên phần tiền xử lý không quá nhiều bước. Với bộ dữ liệu ít ỏi như vậy thì chắc chắn sẽ gây ra nhiều thiếu sót trên thực tế. Tuy nhiên các câu này là do nhóm tự xây dựng nên với các trường hợp nhập nhằng và độ phức tạp tương đối, không sao chép ở bất kỳ nguồn nào. Nên các thành viên trong nhóm tin là tuy là bộ ngữ liệu khó giải quyết được vấn đề phức tạp trên thực tế nhưng sẽ giải quyết được vấn đề được đặt ra trong bài toán này.

Chương 4. GIẢI QUYẾT BÀI TOÁN

1. Framework

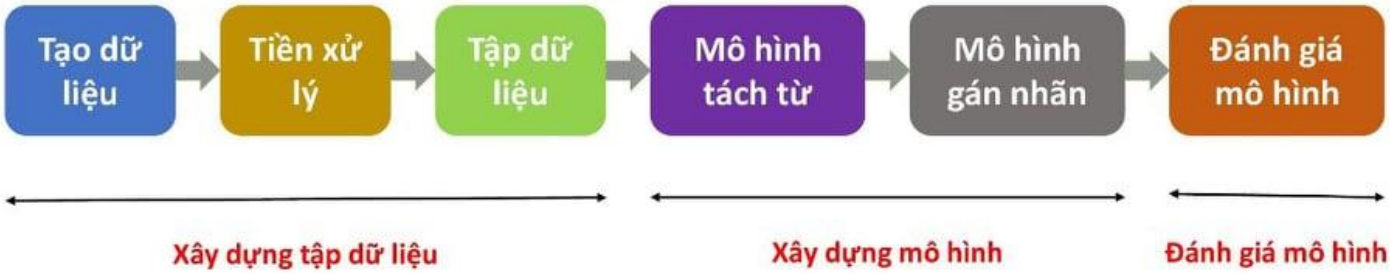
Nhằm đảm bảo việc thiết kế, xây dựng và đánh giá các mô hình gán nhãn từ loại đáp ứng yêu cầu của bài toán đặt ra đi đúng theo mục tiêu được đề ra, chúng tôi đã thiết kế framework tổng quát của quy trình giải bài toán dựa trên mẫu mô hình tổng quát thiết kế và đánh giá các mô hình tách từ và gán nhãn như hình dưới đây. Trong quá trình thực hiện, nhóm luôn bám sát theo framework này nhằm đảm bảo quy trình thiết kế, xây dựng và đánh giá các mô hình được tiến hành một cách logic và đảm bảo đầy đủ các bước theo yêu cầu.

Quy trình này bao gồm ba khâu lớn là xây dựng tập dữ liệu, xây dựng mô hình và đánh giá mô hình.

Xây dựng tập dữ liệu: 40 câu tiếng Việt, nhóm đã thực hiện qua một số khâu tiền xử lý như chuẩn hóa các từ đặc biệt, kiểm tra các từ không trong bộ huấn luyện xuất hiện trong bộ thử nghiệm để xây dựng nên tập dữ liệu cần thiết cho bài toán.

Xây dựng mô hình: Về phần tách từ, mô hình sẽ được tự tay thành viên trong nhóm xây dựng bằng các bước của phương pháp longest matching kết hợp sử dụng một số thư viện như việc tiếp cận một vấn đề theo nhiều hướng khác nhau. Còn phần gán nhãn sẽ được thực hiện bằng phương pháp mô hình Markov ẩn một cách trực tiếp bằng việc sử dụng hàm có sẵn được cung cấp từ thư viện đã cài đặt. Mô hình cũng sẽ được áp dụng smoothing để so sánh với trường hợp không smoothing.

Đánh giá mô hình: Để có cái nhìn khách quan về mô hình, nhóm đã tiến hành đánh giá chất lượng và hiệu quả hoạt động của mô hình này thông qua các phương pháp cơ bản như chỉ số precison, recall, F1 – score và accuracy cho cả phần tách từ và gán nhãn.



Framework tổng quát của quá trình xây dựng và đánh giá mô hình

2. Xây dựng tập dữ liệu

Quy trình xây dựng tập dữ liệu

Trong tập dữ liệu 40 câu tiếng Việt, nhóm đã thực hiện qua một số khâu tiền xử lý như: chuẩn hóa các từ đặc biệt, kiểm tra các từ không trong bộ huấn luyện xuất hiện trong bộ thử nghiệm để xây dựng nên tập dữ liệu cần thiết cho bài toán. Các bước tiền xử lý dữ liệu được thực hiện để xây dựng tập dữ liệu bài hát phục vụ cho bài toán sẽ được trình bày chi tiết ở phần dưới đây.

Các bước tiền xử lý dữ liệu được thực hiện để có thể chuẩn bị được một hoặc nhiều bộ dữ liệu sạch, hoàn chỉnh. Các câu do nhóm tạo ra được hầu hết đều là dữ liệu thô với mục đích là để gắn sát với thực tiễn. Do đó, ta cần thực hiện các bước tiền xử lý dữ liệu trên các câu sau khi đã thu thập dữ liệu xong. Quy trình tiền xử lý dữ liệu có thể bao gồm một hoặc nhiều công đoạn như làm sạch dữ liệu, tích hợp dữ liệu, biến đổi dữ liệu, cân bằng dữ liệu, thu giảm dữ liệu, rời rạc hóa dữ liệu,… Tùy vào các khía cạnh như tính chất dữ liệu, đặc điểm phân bố dữ liệu, các thuộc tính,… của bộ dữ liệu được sử dụng mà chúng ta sẽ lựa chọn các hình thức tiền xử lý dữ liệu thích hợp để thực hiện trong khâu chuẩn bị dữ liệu.

Với 40 câu tiếng Việt sau khi thực hiện qua các khâu tiền xử lý nêu trên đã được lưu trữ thành file csv. Tập dữ liệu này sẽ được chúng tôi sử dụng cho bài toán được đặt ra, đây cũng chính là tập dữ liệu bài hát được sử dụng để phục vụ cho việc tách từ và gán nhãn từ loại trong câu tiếng Việt.

Các bước tiền xử lý xây dựng tập dữ liệu

Chuẩn hóa các từ đặc biệt:

- Mục tiêu của bài toán là gán nhãn từ loại trong câu tiếng Việt. Tuy nhiện, tiếng Việt là một ngôn ngữ phức tạp và không đơn nghĩa nên trong quá trình xây dựng nhóm có thể bắt gặp những trường hợp từ đặc biệt khác so với quy chuẩn tiếng Việt thông thường có thể dẫn đến sai lệch trong kết quả bài toán. Các trường hợp từ đặc biệt được nhóm quy ước là: sai lỗi chính tả, viết tắt, dùng từ mượn, gộp chữ hoa lẫn chữ thường trong một từ, kí tự đặc biệt, những từ theo quy chuẩn riêng. Khi gặp các trường hợp đó nhóm sẽ chuẩn hóa lại các từ theo hệ thống quy tắc thông dụng và không bao gồm chữ in hoa để kết quả bài toán minh bạch nhất có thể.
- Để trực quan hơn về thao tác này, chúng tôi xin đưa ra ví dụ như sau: chẳng hạn trước tiền xử lý sẽ có các câu trong tập dữ liệu rơi vào một trong các trường hợp đặc biệt

Câu tiền xử lý	Trường hợp	Câu mong muốn
trồng tập thật giày	sai lỗi chính tả	chồng tập thật dày
đây là bt mới	viết tắt	đây là bài tập mới
bài tập toán này easy	dùng từ mượn	bài tập toán này dễ
Nó đaNg học bài Này	chữ hoa lẫn chữ thường	nó đang học bài này
đây là một trong những bài tập khó :((	kí tự đặc biệt	đây là một trong những bài tập khó
cây k3o n4`y sắc lắm	những từ theo quy chuẩn riêng	cây kéo này sắc lắm

Ví dụ về các câu chứa trường hợp đặc biệt

Kiểm tra các từ không trong bộ huấn luyện xuất hiện trong bộ thử nghiệm:

- Thực chất đây là việc làm không cần thiết đối với bộ ngữ liệu lớn hơn do các từ xuất hiện trong bộ thử nghiệm hầu như có trong bộ huấn luyện. Ngoài ra, đối với các mô hình lớn thì sẽ có khách phục trường hợp này ví dụ như đối với gán nhãn, một nhãn dành riêng cho từ không xác định sẽ được thêm vào hoặc thuật toán có trong mô hình sẽ tự dự đoán nhãn của các từ đó dựa vào tính toán xác suất những chuỗi nhãn.
- Tuy vậy, mô hình do nhóm xây dựng chỉ giải quyết bài toán quy mô nhỏ, ngữ liệu huấn luyện ít, các công cụ chỉ dùng cho việc tính toán các ngữ liệu sẵn có chứ không được xây dựng từ trước nên khó có thể xử lý các vấn đề phức tạp bên ngoài. Phần tách từ do sử dụng phương pháp phụ thuộc vào bộ từ điển như đã nhắc trong cơ sở lý thuyết nên các từ trong bộ thử nghiệm không có trong từ điển được liệt kê từ bộ huấn luyện chắc chắn sẽ không cho kết quả. Phần gán nhãn thì do các trạng thái phụ thuộc nhau nên khả năng cao nếu sai nhãn và gây nhiễu cho câu, tạo ra chuỗi nhãn sai. Với một mô hình nhỏ thì những sai sót như thế này là không đáng.

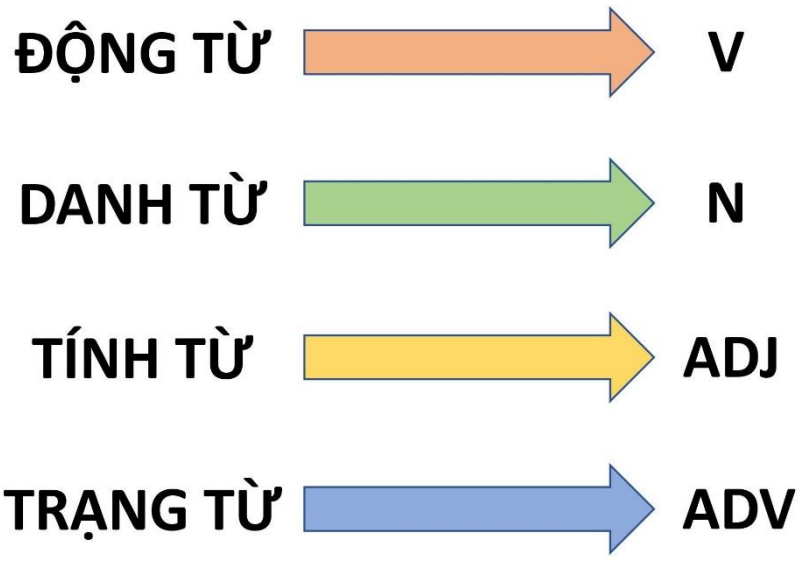
Trường hợp các từ không trong bộ huấn luyện xuất hiện trong bộ thử nghiệm

Các bước tiền xử lý sẽ được nhóm làm thủ công trực tiếp trên bộ dữ liệu do bộ dữ liệu sử dụng trong báo cáo chỉ là bộ dữ liệu nhỏ. Nhóm đã không sử dụng bất kỳ công cụ hỗ trợ nào do việc tìm hiểu, cài đặt và đánh giá kết quả sau khi thực hiện sẽ mất nhiều thời gian và công sức hơn là làm thủ công trên các bộ dữ liệu nhỏ như bộ được dùng trong báo cáo này.

3. Lựa chọn bộ nhãn

Trong phần gán nhãn từ loại, lựa chọn bộ nhãn là một trong những việc cần làm của quá trình này. Việc lựa chọn bộ nhãn hợp lý sẽ phụ thuộc vào các yếu tố: Ngôn ngữ của người sử dụng, độ phức tạp của dữ liệu. Lựa chọn bộ nhãn thích hợp sẽ cho ra kết quả trực quan nhất sau khi thực hiện mô hình, phù hợp với mục tiêu và việc đánh giá của người xem.

Bộ nhãn được lựa chọn không nên quá đơn giản bởi vì có thể sẽ không đáp ứng được đầy đủ yêu cầu bài toán. Đặc biệt khi dữ liệu lớn, việc có quá ít nhãn có thể sẽ không bao quát hết được các trường hợp trong câu.



Nhãn từ loại quá đơn giản

Tùy thuộc vào quan điểm chủ quan của mọi người mà có thể lựa chọn bộ nhãn liên quan tới ngôn ngữ tự nhiên muốn sử dụng. Đối với các ngôn ngữ châu Âu, các lớp từ liên quan đến các khía cạnh hình thái như giới tính, số, trường hợp,… Đối với tiếng Việt, các từ thường được phân loại dựa trên khả năng kết hợp, chức năng cú pháp của chúng và ý nghĩa tổng quan. Nhóm chọn hai tiêu chí đầu tiên, khả năng kết hợp và chức năng cú pháp để lựa chọn nhãn gán. Do đó, bộ nhãn sẽ không chứa thông tin hình thái (số lượng, khía cạnh, thì,…), thông tin phân loại phụ (động từ bắc cầu / nội động từ, động từ theo sau mệnh đề,…) và thông tin ngữ nghĩa (semantic information). Vì vậy, nhóm đã quyết định sử dụng bộ nhãn thuộc nhóm xử lý ngôn ngữ tự nhiên VLSP – bộ nhãn từ loại dành cho tiếng Việt được phát triển bởi tác giả Trần Việt Trung– với một số nhãn được lược bỏ để phù hợp với những thuộc tính được liệt kê trên.

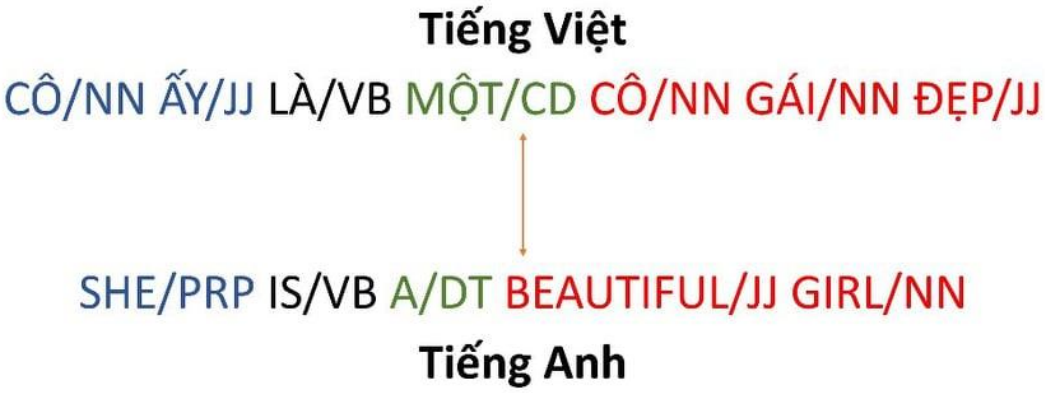
Bộ nhãn gốc		**Bộ nhãn điều chỉnh**	
Nhãn	Từ loại	Nhãn	Từ loại
A	Tính từ	A	Tính từ
C	Liên từ phối hợp	C	Liên từ
S	Liên từ phụ thuộc		
I	Thán từ	I	Thán từ
L	Định từ (những, các, vài, …)	L	Định từ (những, các, vài, …)
M	Số từ	M	Số từ
P	Đại từ	P	Đại từ
R	Trạng từ	R	Trạng từ
E	Giới từ	E	Giới từ

T	Trợ từ, động từ khiếm khuyết	T	Trợ từ, động từ khiếm khuyết
V	Động từ	V	Động từ
X	Từ không xác định	X	Từ không xác định
F	Dấu câu	F	Dấu câu
N	Danh từ	N	Danh từ
Nc	Danh từ chỉ loại		
Ny	Danh từ viết tắt		
Np	Danh từ riêng		
Nu	Danh từ đơn vị		

Bộ nhãn gốc và bộ nhãn đã được nhóm điều chỉnh cho phù hợp với mục đích sử dụng

Trong quá trình xây dựng mô hình, thật ra nhóm cũng đã thử nghiệm trên bộ nhãn khác, cụ thể là bộ Penn Treebank. Tuy nhiên, Penn Treebank lại là bộ nhãn chuyên dùng cho tiếng Anh nên khi áp dụng vào các câu tiếng Việt trong mô hình đã nảy sinh vấn đề: cấu trúc ngữ pháp của hai ngôn ngữ là không hề giống nhau nên trật tự trong câu có thể đảo vị trí khi dịch từ ngôn ngữ này sang ngôn ngữ kia, tiếng Anh đơn nghĩa nên một từ tiếng Anh khi dịch sang tiếng Việt sẽ có thể trở thành từ ghép gồm các từ đơn có nghĩa khác nhau nên chuỗi nhãn sẽ bị kéo dài ra hơn so với dự tính. Ngoài ra, nhãn từ có thể sẽ bị đổi khác với mục đích ban đầu do sự không đồng nhất từ loại trong bộ từ điển khi dùng bộ Penn Treebank. Tuy việc sử dụng bộ Penn Treebank không gây

sai sót cho gán nhãn câu tiếng Việt nếu biết cách thực hiện. Nhưng đây là một việc vô cùng rối rắm và có thể gây ra nhầm lẫn do yêu cầu kiến thức ngôn ngữ học cao.



Ví dụ về việc rối rắm khi sử dụng bộ nhãn Penn Treebank trong câu tiếng Việt so với câu tiếng Anh

4. Xây dựng mô hình tách từ longest matching

Lý do chọn phương pháp

Do đây là một trong các phương pháp tiếp cận dựa trên từ điển, các từ trong câu sẽ được so khớp với các từ có trong bộ từ điển được tạo ra và sẽ được tách thành một từ nếu khớp với từ có trong từ điển. Hiện nay thì longest matching được xem là phương pháp hiệu quả nhất trong hướng tiếp cận này.

Longest matching không phải là một phương pháp học máy hiện đại, nên sẽ không linh hoạt, độ chính xác sẽ phụ thuộc vào độ chính xác và đầy đủ của từ điển. Việc xây dựng một bộ từ điển hoàn chỉnh là vấn đề cốt lõi của hướng tiếp cận này, bộ từ điển hiện nay đã tương đối đầy đủ đem lại kết quả khá khả quan cho phương pháp này với độ chính xác cao trong việc tách từ.

Việc tách từ là bước quan trọng và được xem như tiền xử lý của gán nhãn nên nhóm đã lựa chọn phương pháp này để mang lại độ chính xác cao nhất. Các phương

pháp áp dụng máy học có thể sẽ tốn ít công sức và dễ bảo trì hơn. Tuy nhiên, với bộ dữ liệu nhỏ, các từ được sử dụng lại nhiều lần để tạo trường hợp nhập nhằng nên việc tự xây dựng từ điển không gây nhiều khó khăn.

Xây dựng từ điển

Từ điển dùng trong mô hình tách từ này là một tập hợp chứa các từ xuất hiện trong bộ dữ liệu. Ta chỉ cần liệt kê các từ cần tách so với các từ còn lại và gán số thứ tự cho chúng

Ví dụ: Ta có các câu sau:

- đây là bài tập mới
- đây là một trong những bài tập khó
- nó đang làm bài tập
- bài tập toán này dễ

Dictionary sẽ được tạo sẽ có cấu trúc như sau:

- dict = {"đây": 0, "là": 1, "bài tập": 2, "mới":3, "một": 4, "trong": 5, "những": 6, "khó": 7, "nó": 8, "đang": 9, "làm": 10, "toán": 11, "này": 12, "dễ": 13}

Quá trình thực hiện

Mô hình tách từ sẽ được xây dựng thủ công dựa trên thuật toán đã được nhắc đến trong phần cơ sở lý thuyết mà không sử dụng bất kỳ các công cụ trợ giúp nào.

Việc tách từ trong từng câu được dựa trên ký tự khoảng trắng và từ ghép. Các từ đơn được ngăn cách với nhau bằng khoảng trắng sẽ được giữ nguyên. Các từ ghép sẽ được phân biệt bằng cách thay khoảng trắng giữa các từ đơn trong từ ghép đó bằng dấu "_".

Ví dụ: câu *đây là một trong những bài tập khó* sau khi thực hiện quá trình tách từ sẽ thành ['đây ', 'là ', 'một ', 'trong ', 'những ', 'bài_tập ', 'khó']

Ngoài ra mô hình tách từ còn sử dụng một số thư viện là pyvi và underthesea để thử nghiệm phương pháp học máy trên bộ dữ liệu sẵn có.

5. Xây dựng mô hình gán nhãn Hidden Markov Model

Xây dựng tập huấn luyện

Tập huấn luyện sử dụng cho gán nhãn sẽ được lấy trực tiếp từ bộ dữ liệu sau khi được tách từ ở phần trên, bộ dữ liệu tách từ phải đảm bảo độ chính xác cao nhất vì nếu có sai sót trong phần tách từ thì sẽ làm ảnh hưởng độ chính xác của mô hình gán nhãn một cách trực tiếp.

Các nhãn sẽ được gán thủ công vào từng từ trong các câu, 32 câu sẽ được dùng để huấn luyện mô hình còn 8 câu còn lại sẽ được làm bộ test. Ngoài ra, 8 câu test đấy còn được gán nhãn đúng hoàn toàn để làm bộ gold so sánh với kết quả cuối cùng do mô hình thực hiện.

Xây dựng mô hình

Mô hình gán nhãn bằng Hidden Markov Model sẽ được xây dựng bằng NLTK, một nền tảng xây dựng các chương trình trên ngôn ngữ lập trình Python để hoạt động với dữ liệu ngôn ngữ của con người, chuyên cung cấp các giao diện dễ sử dụng cho hơn 50 tài nguyên ngữ liệu và từ vựng như WordNet, cùng với một bộ thư viện xử lý văn bản để phân loại, tách từ, stemming, gán nhãn, phân tích cú pháp và lập luận ngữ nghĩa.

Module được sử dụng là nltk.tag.hmm và class được sử dụng trong module là HiddenMarkovModelTrainer, hàm này sẽ tạo một trình đào tạo để tạo ra một HMM với các trạng thái nhất định và ký hiệu đầu ra. Method được dùng trong class đó là train_supervised: huấn luyện có giám sát tối đa hóa xác suất chung của chuỗi nhãn và trạng thái. Điều này được thực hiện thông qua việc thu thập tần số chuyển đổi giữa các trạng thái, quan sát ký hiệu khi ở trong mỗi trạng thái và trạng thái nào bắt đầu một câu. Các phân bố tần số này sau đó được chuẩn hóa thành các ước lượng xác suất, có thể được áp dụng được smoothing.

Smoothing

Phương pháp smoothing sẽ được áp dụng vào mô hình để so sánh với kết quả không smoothing. Việc này sẽ cho ta thấy được cái nhìn trực quan hơn về mô hình gán nhãn.

Phương pháp smoothing được sử dụng: Laplace smoothing. Hàm smoothing sẽ được import từ nltk.probability tương ứng với phương pháp trên là LaplaceProbDist và sẽ được gán vào tham số estimator của method train_supervised.

6. Đánh giá mô hình

Phương pháp

Để đánh giá cho cả hai bài toán tách từ và gán nhãn, nhóm đã sử dụng các phương pháp đánh giá chuyên dùng cho các mô hình học máy như accuracy, Precision, Recall và F1 – score.

Tách từ

Accuracy: Cách đơn giản và hay được sử dụng nhất là accuracy (độ chính xác). Đối với bài toán tách từ, ta sẽ so sánh bộ thử nghiệm sau khi được tách từ và bộ tách từ thủ công. Kết quả sẽ là tỉ lệ giữa các từ được tách của bộ thử nghiệm đúng với các từ được tách trong bộ thủ công và mẫu sẽ là tổng số từ có trong bộ tách thủ công được chắc chắn là đúng. Công thức tính có thể được khái quát là: Acc = n / N, trong đó:

- n là số lượng từ tách đúng trong tập thử nghiệm
- N là số lượng từ trong tập thủ công
- $n \leq N$

Để rõ hơn, ta có hai ví dụ sau:

Ví dụ 1:

Tập tách từ thủ công	Tập tách từ thử nghiệm
Một cậu học_sinh	Một cậu học sinh
Nó đang làm bài_tập	Nó đang làm bài_tập
Ông_già đi nhanh quá	Ông già đi nhanh quá

Ta có thể thấy:

- Số từ tách đúng trong bộ thử nghiệm: **n** = 9
- Số từ trong bộ thủ công: **N** = 11

  ⇨ accuracy = 9 / 11 = 0.82

Ví dụ 2:

Tập tách từ thủ công	Tập tách từ thử nghiệm
Một cậu học sinh	Một cậu học_sinh
Nó đang làm bài_tập	Nó đang làm bài_tập
Ông_già đi nhanh quá	Ông già đi nhanh quá

Ta có thể thấy:

- Số từ tách đúng trong bộ thử nghiệm: **n** = 9
- Số từ trong bộ thủ công: **N** = 12

  ⇨ accuracy = 9 / 12 = 0.75

Recall và precision: Trong bài toán phân lớp, để tìm được recall và precision do thì trước hết cần phải có các tham số đánh giá là: True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN):

- True positive (TP): Kết quả dự đoán đúng và kết quả làm chuẩn đúng.
- False positive (FP): Kết quả dự đoán đúng và kết quả làm chuẩn sai.
- True negative (TN): Kết quả dự đoán sai và kết quả làm chuẩn sai.
- False negative (FN): Kết quả dự đoán sai và kết quả làm chuẩn đúng.

Từ các tham số trên, ta có định nghĩa:

- Precision: Xác suất trường hợp positive trong tổng số trường hợp positive được dự đoán. Ở đây Mẫu số là tổng các trường hợp positive được dự đoán trong mô hình từ toàn bộ tập dữ liệu đã cho. Precision có thể được xác định nôm na là "mô hình đúng bao nhiêu khi nó là đúng". Precision được tính như sau:

$$\text{Precision} = \frac{TP}{TP+FP}$$

- Recall: Xác suất trường hợp positive trong tổng số trường hợp thực sự là positive. Do đó, mẫu số (TP + FN) là số lượng thực tế các trường hợp positive có trong tập dữ liệu. Có thể hiểu recall như là "mô hình đã bỏ lỡ bao nhiêu cái đúng khi nó hiển thị cái đúng".

$$\text{Recall} = \frac{TP}{TP+FN}$$

Trong mô hình tách từ tiếng Việt của báo cáo này, để tính precision và recall nhóm sẽ dựa vào số từ ghép xuất hiện trong tập tách thủ công và tập thử nghiệm. Tham số true positive là số các từ ghép tách được trong bộ thử nghiệm trùng khớp với vị trị số từ ghép trong bộ thủ công. False positive là khi từ ghép có trong bộ thử nghiệm nhưng không có tại vị trí đó trong bộ thủ công. Còn false negative là số từ ghép có trong bộ thủ công nhưng tại các vị trí đó trong bộ thử nghiệm không phải là từ ghép tương

ứng. Nhóm sẽ tiếp tục ví dụ 2 ở phần tính accuracy để minh họa cho cách tính precision và recall cho mô hình tách từ.

Ví dụ 2:

Tập tách từ thủ công	Tập tách từ thử nghiệm
Một cậu học sinh	Một cậu học_sinh
Nó đang làm bài_tập	Nó đang làm bài_tập
Ông_già đi nhanh quá	Ông già đi nhanh quá

Các tham số đánh giá:

- True positive: 1
- False positive: 1
- False negative: 1

⇨ Precision = $\frac{1}{1+1}$ = 0.5

⇨ Recall = $\frac{1}{1+1}$ = 0.5

F1 – score: Là harmonic mean của precision và recall, thể hiện sự cân bằng giữa hai đại lượng. F1 – score được tính như sau:

$$F1 = 2\frac{1}{\frac{1}{precision}+\frac{1}{recall}} = 2\frac{precision*recall}{precision+recall}$$

Chỉ số F1 – score của ví dụ trên là: $2\frac{0.5*0.5}{0.5+0.5}$ = 0.5

Đối với phần tách từ, nhóm sẽ xây dựng thủ công các thuật toán tương ứng trong mô hình để tính các tham số đánh giá trên.

Accuracy: Để đánh giá một mô hình gán nhãn dựa vào accuracy, đầu tiên ta sẽ tạo một tập gold với các câu đã được gán nhãn thủ công mà ta đã chắc chắn là đúng và một bộ dữ liệu thử nghiệm đã được gán nhãn. Sau đó tính tỉ lệ giữa số nhãn được dự đoán đúng trong bộ thử nghiệm với tổng số nhãn trong tập gold. Công thức tính có thể được khái quát là: Acc = n / N, trong đó:

- n là số lượng nhãn dự đoán đúng
- N là số lượng nhãn trong tập gold
- $n \leq N$

Ví dụ: cho bộ ngữ liệu cần gán nhãn với các câu như sau:

- cây kéo này sắc lắm
- bọn trẻ đang kéo nhau
- cái bàn này vững lắm

Ta có kết quả gán nhãn của tập gold so với bộ ngữ liệu trên trong bảng:

Bộ gold	Bộ thử nghiệm
cây/N kéo/N này/P sắc/A lắm/R	cây/N kéo/V này/P sắc/A lắm/R
bọn/N trẻ/N đang/R kéo/V nhau/N	bọn/N trẻ/N đang/R kéo/N nhau/N
cái/N bàn/N này/P vững/A lắm/R	cái/N bàn/V này/P vững/A lắm/R

Dựa vào bảng ta thấy được:

- Số nhãn đúng trong bộ thử nghiệm: n = 12
- Số nhãn trong bộ gold: N = 15

⇨ accuracy = 12 / 15 = 0.8

Recall và precision: Trong mô hình gán nhãn, mỗi từ chỉ có duy nhất một nhãn nên khi so sánh với bộ gold thì false positive và true negative luôn bằng 0. Do đó, việc

tính precision và recall cho cả mô hình là vô nghĩa vì cả hai đều sẽ bằng với accuracy. Nhưng hai phương pháp trên có thể áp dụng cho từng nhãn xuất hiện trong mô hình như tính chỉ số precision và recall cho nhãn N. Tham số true positive có thể xem như số lần mà nhãn đó trong bộ thử nghiệm và bộ gold trùng nhau cho cùng một từ. False positive là khi nhãn đó bị gán nhầm sang cho từ có nhãn khác trong bộ thử nghiệm. Còn false negative là khi mô hình dư đoán sai nhãn cho từ trong bộ thử nghiệm so với nhãn đang xét trong bộ gold. Để dễ hình dung, nhóm xin dùng lại ví dụ ở phần tính accuracy trong gán nhãn với mục đích là tính precision và recall cho nhãn N.

Ta có 2 chuỗi nhãn từ bộ gold và bộ thử nghiệm trên như sau:

- Bộ gold: ['N', 'N', 'P', 'A', 'R', 'N', 'N', 'R', 'V', 'N', 'N', 'N', 'P', 'A', 'R']
- Bộ test: ['N', 'V', 'P', 'A', 'R', 'N', 'N', 'R', 'N', 'N', 'N', 'V', 'P', 'A', 'R']

Các tham số đánh giá dựa vào tập liệu:

- True positive: 5
- False positive: 1
- False negative: 2

⇨ Precision = $\frac{5}{5+1}$ = 0.83

⇨ Recall = $\frac{5}{5+2}$ = 0.71

F1 – score: Chỉ số F1 – score của gán nhãn cũng sẽ được tính như phần tách từ. Nếu xét tiếp ví dụ trên thì chỉ số F1 – score của nhãn N là: $2\frac{0.83*0.71}{0.83+0.71}$ = 0.77

Tất cả các phương pháp đánh giá trên trong phần gán nhãn sẽ được sử dụng hàm classification_report của thư viện scikit-learn để đánh giá.

Chương 5. CÀI ĐẶT THỬ NGHIỆM

1. Thiết kế chương trình cài đặt

Ngôn ngữ lập trình

Python ra đời năm 1991, và là một ngôn ngữ thông dịch. Trải qua gần 30 năm phát triển, Python là một trong những ngôn ngữ được sử dụng nhiều nhất trong dạy lập trình và nghiên cứu khoa học. Rất nhiều trường đại học sử dụng Python để dạy về lập trình cho các sinh viên ngành Khoa Học Máy Tính. Rất nhiều công ty lớn sử dụng Python để xây dựng hệ thống như Google, Youtube, Instagram, Dropbox, Atlassian... Python là một ngữ sử dụng được cho nhiều mô hình lập trình, đơn giản khi học và sử dụng. Và đặc biệt là nó cung cấp cho các lập trình viên rất nhiều thư viện tuyệt vời về máy học, thị giác máy tính, xử lý ngôn ngữ tự nhiên,… như Scikitlearn, Tensorflow, OpenCV và đặc biệt là nltk (The Natural Language Toolkit) – một package của Python về Xử lý ngôn ngữ tự nhiên. Vì thế, nhóm đã chọn Python làm ngôn ngữ để cài đặt chương trình thực nghiệm.

Môi trường lập trình

Jupyter Notebook là một ứng dụng web mã nguồn mở cho phép chạy Interactive Python (hay IPython), bạn có thể đưa cả code Python và các thành phần văn bản phức tạp như hình ảnh, công thức, video, biểu thức... vào trong cùng một file giúp cho việc trình bày trở lên dễ hiểu, giống như một file trình chiếu nhưng lại có thể thực hiện chạy code tương tác trên đó, cốt lõi của việc này chính là Markdown. Các file "notebook" này có thể được chia sẻ với mọi người và có thể thực hiện lại các công đoạn một cách nhanh chóng và chính xác như những gì bạn đã làm trong quá trình tạo ra file.

Google colab hay Colaboratory là một sản phẩm miễn phí do Google nghiên cứu và dựa trên Jupyter. Colab là một công cụ tuyệt vời cho cả người mới bắt đầu và người dùng nâng cao. Hầu hết tất cả các thư viện quan trọng đều được cài đặt sẵn, vì vậy ta không cần phải cài đặt từng cái một. Các file của Colab được lưu trữ trong google drive của người dùng, vì vậy ai cũng có thể truy cập chúng từ bất kỳ đâu. Google colab cũng cho phép chia sẻ file với người dùng khác mà không cần tải xuống, đây được xem như

là tính năng tốt nhất đối với nhiều người. Ngoài ra, sản phẩm này còn cung cấp GPU và TPU miễn phí cho công việc và điều đó làm cho Google colab trở nên lý tưởng cho các dự án học sâu và máy học. Để sử dụng Google Colab, ta không phải cài đặt bất cứ thứ gì mà có thể truy cập trực tiếp vào trang web và bắt đầu sử dụng.

Nhóm chọn Jupyter notebook để có thể thử nghiệm code dễ dàng và Google colab để thuận tiện cho việc chia sẻ file. Cả hai thực chất khá giống nhau, đều trình bày kết quả một cách trực quan hơn là dùng các IDE.

Các thư viện sử dụng

Sau đây là các thư viện và tác dụng của chúng trong chương trình cài đặt của nhóm:

STT	Tên thư viện	Mục đích sử dụng	Các biến/hàm được sử dụng	Chức năng
1	Pandas	Thao tác với file	read_csv()	Đọc các giá trị được phân tách bằng dấu phẩy trong file csv vào một DataFrame.
			tolist()	Trả về list của các giá trị
2	nltk	Xây dựng mô hình gán nhãn	HiddenMarkovModelTrainer	Tạo một trình huấn luyện HMM
			train_supervised()	Huấn luyện có giám sát tối đa hóa xác suất chung của chuỗi ký hiệu và trạng thái

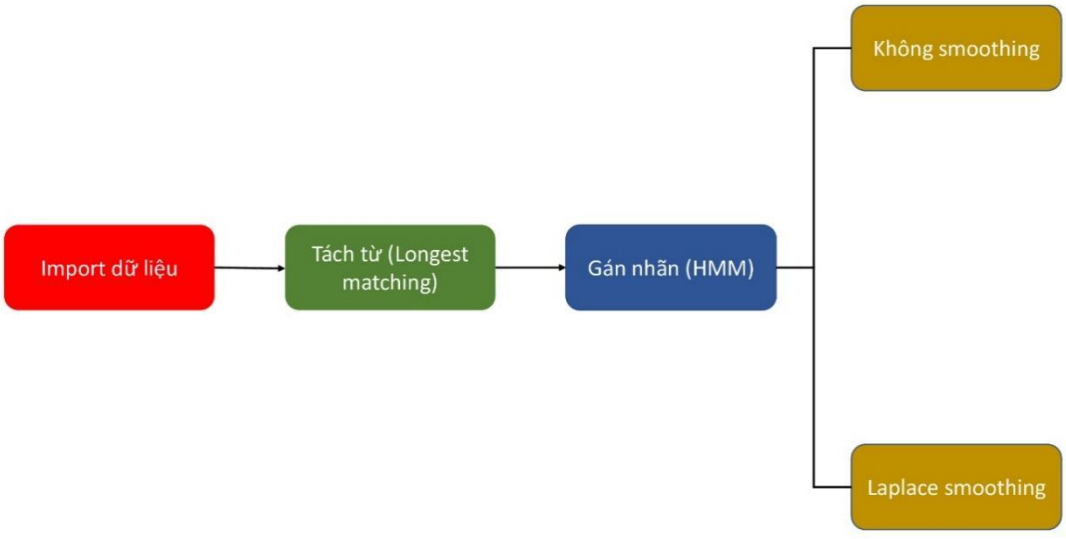
3	scikit-learn	Đánh giá mô hình	classification_report()	Xây dựng một báo cáo hiển thị các chỉ số đánh giá chính.
4	pyvi*	Tách từ	ViTokenizer()	Tách từ sử dụng phương pháp Conditional Random Field.
5	underthesea*	Tách từ	word_tokenize()	Tách từ sử dụng phương pháp Conditional Random Field.

Các thư viện được sử dụng trong chương trình cài đặt

**Việc sử dụng thư viện pyvi và underthesea là hoàn toàn tự chọn (optional). Phương pháp chính vẫn là sử dụng longest matching. Nhóm chỉ cài đặt vào để so sánh hiệu suất các phương pháp với nhau.*

Các trường hợp cài đặt

Sau đây là các phương pháp có trong mô hình do nhóm cài đặt thử nghiệm:



Các phương pháp có trong mô hình

STT	Các phương pháp
1	Tách từ (Longest matching) + Gán nhãn (không smoothing)
2	Tách từ (Longest matching) + Gán nhãn (laplace smoothing)

Các phương pháp có trong mô hình

Giao diện chương trình cài đặt

Giao diện web, do là dùng Jupyter Notebook và Google Colab.

Thiết kế theo hướng module hóa nên có thể thay thế hoặc bổ sung thêm các chức năng và các cách cài đặt hàm.

Cấu trúc chương trình cài đặt

Dưới đây là sơ đồ cấu trúc chương trình cài đặt

Trong sơ đồ dưới đây, quy trình sẽ đi theo trình tự là: Trong sơ đồ dưới đây, quy trình sẽ đi theo trình tự là: “Tách từ” và sau đó mới tới “Gán nhãn”. Trong “Tách từ”, các bước sẽ được thực hiện tuần tự từ trên xuống dưới và “Gán nhãn” thì cũng tương tự như vậy.

Sơ đồ cấu trúc chương trình cài đặt

Quy trình thiết kế chương trình cài đặt

Với sơ đồ thiết kế như trên thì chúng ta sẽ viết các hàm, mỗi hàm sẽ thực hiện những chức năng tương ứng của chúng đến lúc hoàn tất chương trình. Mục đích viết hàm là để có thể dễ dàng thay thế sửa chữa trong suốt quá trình. Quá trình diễn ra như sau:

- Đầu tiên ở phần tách từ, ta sẽ tạo các bộ dữ liệu cần thiết, một bộ dữ liệu cho tách từ và một bộ đã được tách thủ công dùng để đánh giá kết quả. Hai bộ dữ liệu này sẽ được lưu vào hai biến và được đọc cùng một file csv. Hàm đọc file sẽ được lấy từ thư viện pandas.
- Tiếp theo, để có thể tách từ sử dụng phương pháp longest matching, đầu tiên, một hàm sẽ được viết với các thuật toán tương ứng để so khớp các từ có trong bộ dữ liệu cần tách và bộ từ điển. Đồng thời, trong bước này, một bộ từ điển sẽ được tạo ra gồm những từ cần tách.
- Sau khi tách từ hoàn tất, các hàm thống kê sẽ được thiết kế để đếm tổng từ tách được, số từ đơn và số từ ghép cùng với việc thống kê từ ghép.
- Để đánh giá phần tách từ, ba hàm sẽ được viết. Hàm đầu tiên sẽ đếm số từ tách đúng. Hàm thứ hai có nhiệm vụ thống kê các tham số đánh giá như TP, FP, FN. Hàm thứ ba sẽ tính toán các chỉ số đánh giá là accuracy, precision, recall, F1 – score.
- Phần thứ hai là gán nhãn. Để tạo bộ dữ liệu huấn luyện, nhóm sẽ gán nhãn thủ công 32 câu đầu tiên của dữ liệu đã qua tách từ, 8 câu còn lại sẽ được dùng làm bộ thử nghiệm và đồng thời một bộ gold sẽ gồm 8 câu thử nghiệm đã được gán nhãn chính xác.
- Từ bộ dữ liệu, một hàm huấn luyện và gán nhãn sẽ được tạo ra để gán nhãn không sử dụng smoothing và có smoothing Laplace.
- Cuối cùng, để đánh giá tách từ, ta sẽ có một hàm chuyển chuỗi nhãn của các từ được gán vào một list. Chuỗi nhãn của bộ gold và các chuỗi nhãn sau khi được tạo bằng mô hình HMM sẽ được chuyển về các list tương ứng với mục đích là đánh giá chỉ số accuracy của toàn mô hình, precision và recall cho từng nhãn. Hàm tính các chỉ số trên sẽ được trích từ thư viện scikit – learn.

File dữ liệu và file chương trình của mô hình sẽ được gửi kèm trong báo cáo hoặc có thể click vào đây và upload file dataset.csv gửi kèm bào cáo để xem.

2. Cấu hình của máy tính được sử dụng để thực nghiệm

Sau đây là thông tin cấu hình của máy tính được sử dụng để chạy thực nghiệm mô hình trong báo cáo:

Trường thông tin	Giá trị
Operating System	Windows 10 Home Single Language Version: 10.0.19043 Build 19043
OS Manufacturer	Microsoft Corporation
Language	English
System Manufacturer	Micro-Star International Co., Ltd.
System Type	64-bit operating system, x64-based processor
System Model	GF63 Thin 10SCXR
BIOS Version	3.2
BIOS mode	UEFI
Processor	Processor Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 2496 Mhz, 4 Core(s), 8 Logical Processor(s)
Memory	8192MB RAM

Thông tin cấu hình của máy tính được sử dụng để chạy thực nghiệm mô hình cài đặt

3. Thiết kế quy trình đánh giá mô hình được cài đặt thử nghiệm

Tiêu chí đánh giá

Với việc thực nghiệm để đánh giá mô hình, nhóm đã đặt ra tiêu chí đánh giá tách từ và gán nhãn như các phương pháp được nêu ở phần 4. Nhóm cũng đã thống kê thực nghiệm bằng cách đếm các chỉ số trả về. Sau đó ghi lại kết quả của từng phương pháp thực nghiệm để so sánh.

Nhắc lại về các phương pháp đánh giá: chỉ số accuracy thể hiện số kết quả đúng so với tổng kết quả, precision được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive (TP + FP), recall được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive (TP + FN), F1-score là trung bình điều hòa của precision và recall.

Quy trình đánh giá

Sau khi có được kết quả, nhóm sẽ tổng hợp lại, và phân tích khác nhau dựa theo các chỉ số khác nhau, cụ thể nhóm chia ra các bảng khác nhau để phân tích và so sánh theo từng đặc tính. Cụ thể tên của các bảng được nhóm chia ra là: “Bảng thể hiện tổng từ tách đúng của longest matching so với thư viện”, “Bảng thể hiện các chỉ số đánh giá của tách từ”, “Bảng thể hiện tổng số từ gán nhãn đúng”. “Bảng thể hiện các chỉ số đánh giá cho từng nhãn xuất hiện”.

Sau khi có được các bảng thực nghiệm do nhóm tổng hợp lại và chia thành các bảng riêng biệt thì nhóm sẽ bắt đầu đánh giá dựa vào chỉ số theo từng bảng. Nhóm sẽ đánh giá dựa theo tiêu chí phương pháp thực nghiệm nào đạt kết quả tốt nhất/cao nhất, phương pháp thực nghiệm nào đạt kết quả kém nhất/thấp nhất. Qua đó sẽ rút ra được những kết luận cụ thể cho từng bảng riêng biệt, từ đó phân tích ý nghĩa của việc chênh lệch cao thấp đó, và giải thích lý do có được sự cao thấp, khác biệt.

Thực nghiệm đánh giá

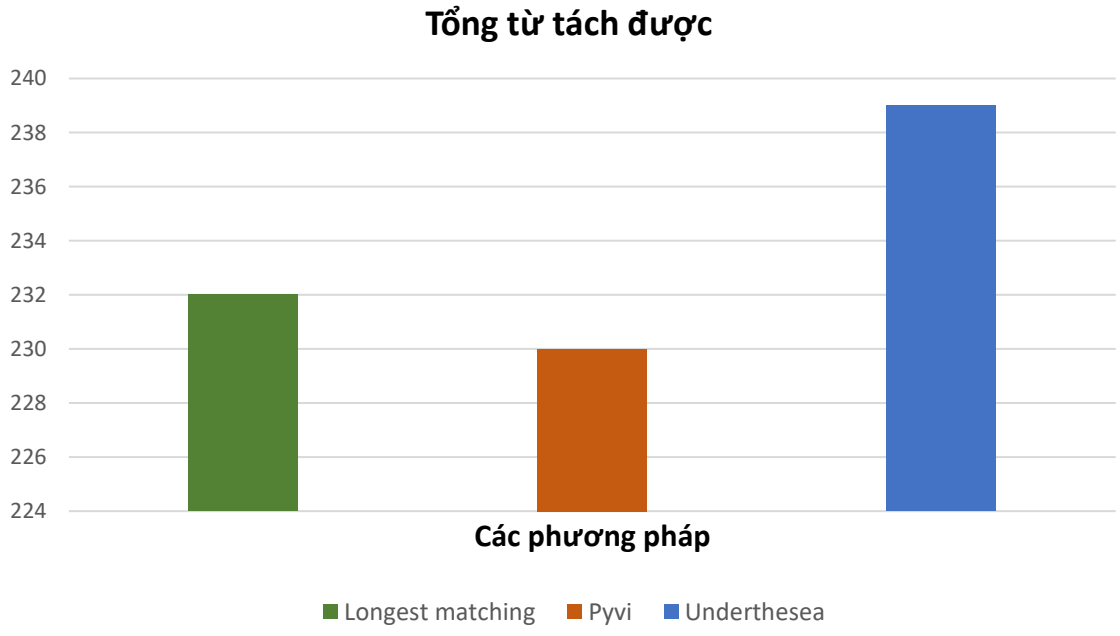
Trong phần thực nghiệm của đề tài được nhóm tổng hợp, thống kê và phân tích, ghi nhận những số liệu thực nghiệm. Các kết quả thực nghiệm do nhóm cài đặt trên máy tính đã được nhóm ghi lại chi tiết và đầy đủ, từ đó để dễ dàng quan sát, nhằm làm cơ sở để đưa ra những kết luận về việc so sánh, từ đó lựa chọn ra được đâu là phương pháp tách từ tốt nhất, mô hình gán nhãn tốt nhất, hiệu quả nhất cho bài toán mà nhóm xử lý.

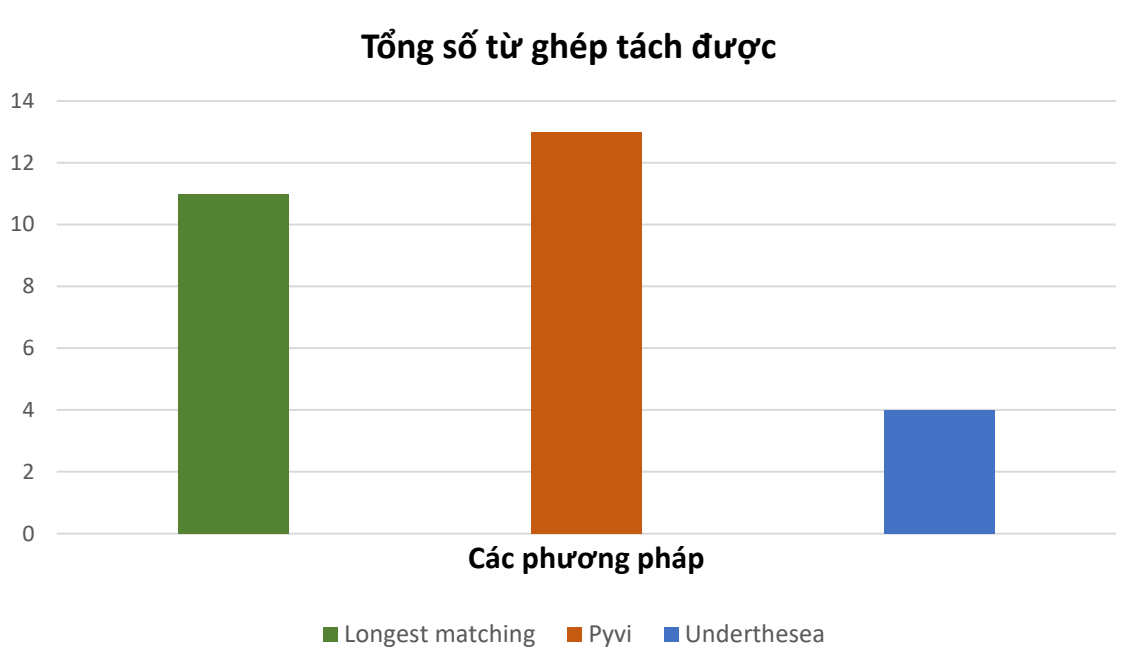
Trải qua quy trình đánh giá đã giúp cho một số thành viên trong nhóm chúng tôi hiểu hơn về cách thức, các bước tiến hành khi đánh giá một phương pháp thực nghiệm nào đó, điều này giúp nâng cao kiến thức của từng thành viên trong nhóm.

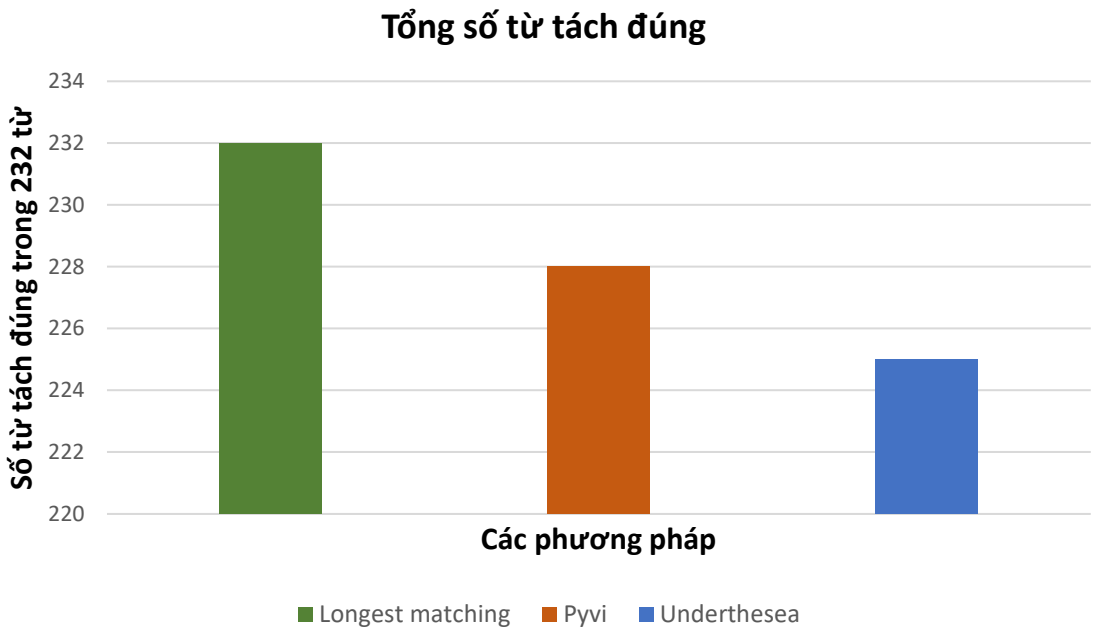
4. Kết quả thực nghiệm

Khi tiến hành chạy thực nghiệm chương trình do nhóm cài đặt trên máy tính có cấu hình nêu trên, chúng tôi thu được kết quả thực nghiệm như dưới đây. Chúng tôi thống kê kết quả theo từng độ đo đánh giá để dễ dàng quan sát và đối chiếu các phương pháp đã được đề xuất, từ đó có thể so sánh và rút ra kết luận về các phương pháp. Sau đây là các biểu đồ thống kê các độ đo đánh giá của các phương pháp.

Tổng từ tách đúng của longest matching so với thư viện




Tổng số từ ghép tách được
14
12
10
8
6
4
2
0
Các phương pháp
Longest matching
Pyvi
Underthesea


Tổng số từ tách đúng
Số từ tách đúng trong 232 từ
234
232
230
228
226
224
222
220
Các phương pháp
Longest matching
Pyvi
Underthesea

Phương pháp	Tổng từ tách được	Tổng từ ghép tách được	Tổng từ tách đúng
Longest matching	232	11	232
Thư viện pyvi	230	13	228
Thư viện underthesea	239	4	225

Bảng thống kê số lượng trong phần tách từ

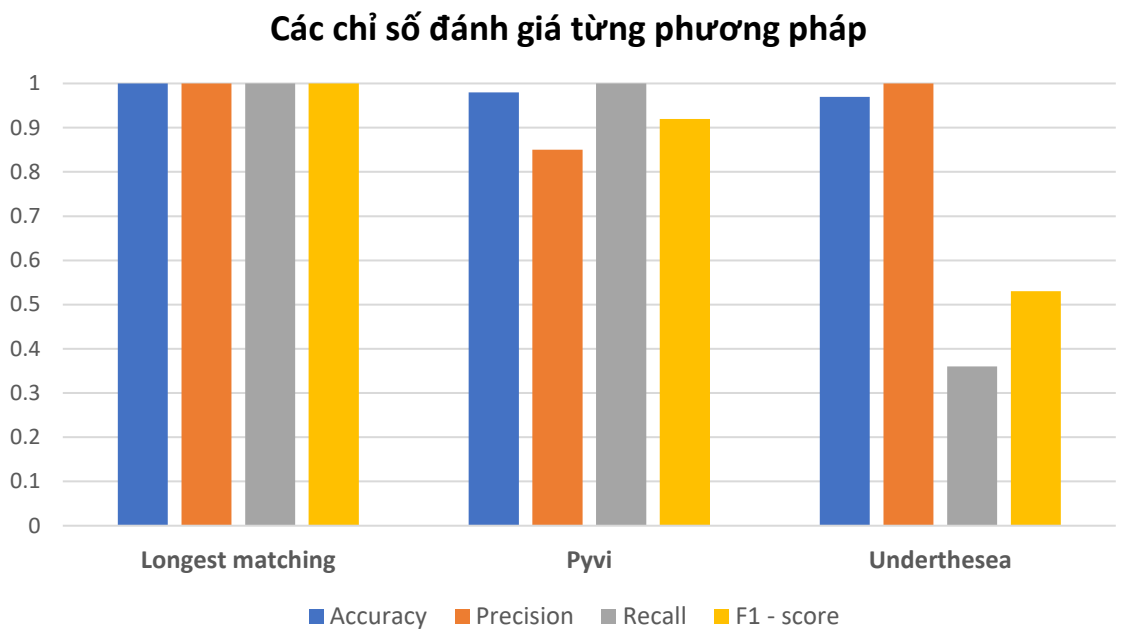
Thông qua các biểu đồ "Tổng số từ tách được", "Tổng số từ ghép tách được", "Tổng số từ tách đúng", ta có thể nhận thấy rằng:

- Phương pháp longest matching tách được 232 từ trong đó có 11 từ ghép và có 232 từ tách đúng.
- Sử dụng thư viện pyvi tách được 230 từ trong đó có 13 từ ghép và có 228 từ tách đúng.
- Sử dụng thư viện underthesea: tách được 239 từ trong đó có 4 từ ghép và có 225 từ tách đúng.

Thuật toán longest matching có số từ tách đúng nhiều nhất (232/232) còn thư viện underthesea có số từ tách đúng ít nhất (225/232). Ta có thể nói longest matching là phương pháp tốt nhất.

Nhưng để có thể đánh giá và đưa ra nhận xét khách quan hơn cho phần từ, ta sẻ xét tiếp phần sau khi có các chỉ số đánh giá đầy đủ.

Các chỉ số đánh giá của tách từ



Phương pháp	Accuracy	Precision	Recall	F1 - score
Longest matching	1	1	1	1
Thư viện pyvi	0.98	0.85	1	0.92
Thư viện underthesea	0.97	1	0.36	0.53

Bảng các chỉ số đánh giá của tách từ

Thông qua các biểu đồ "Các chỉ số đánh giá từng phương pháp", ta có thể nhận thấy rằng:

- Tách từ bằng phương pháp longest matching có chỉ số accuracy = 1, precision = 1, recall = 1, F1 – score = 1

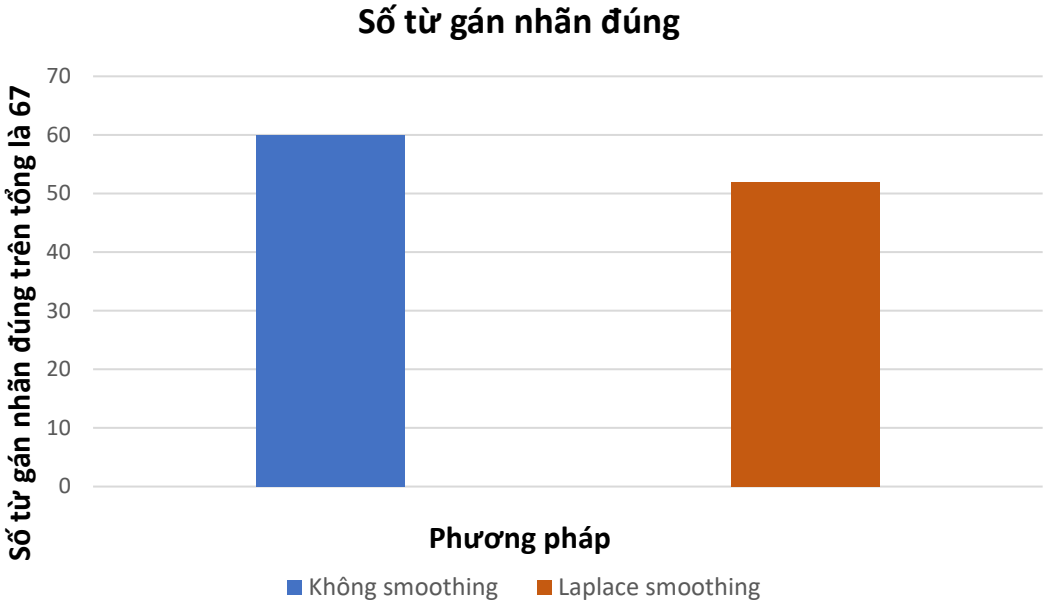
- Tách từ bằng thư viện pyvi có chỉ số accuracy = 0.98, precision = 0.85, recall = 1, F1 – score = 0.92
- Tách từ bằng thư viện underthesea có chỉ số accuracy = 0.97, precision = 1, recall = 0.36, F1 – score = 0.53

Nhóm có một số nhận xét:

- Cả 3 phương pháp trên đều có độ chính xác cao (≥ 0.97) do việc tách từ được thực hiện trong một bộ dữ liệu nhỏ, số lượng từ ghép không nhiều chỉ với tổng là 11 từ và chỉ có 2 từ ghép xuất hiện trong bộ dữ liệu là (bài_tập và bài_học).
- Thư viện pyvi trong lúc tách từ đã tách dư số lượng từ ghép (13) chứ không tách thiếu từ ghép dẫn đến việc tăng FP nhưng không tăng FN. Tuy vậy, số lượng từ ghép tách dư chỉ là 2 nên kết quả precision vẫn cao còn recall vẫn bằng 1 dẫn đến F1 – score cao.
- Thư viện underthesea trong lúc tách từ đã tách thiếu số lượng từ ghép (4) nhưng không sai dẫn đến việc tăng FN nhưng không tăng FP. Số lượng từ ghép tách thiếu là 7 nên kết quả precision vẫn bằng 1 còn recall thì lại thấp dẫn đến F1 – score chỉ ở tầm trung bình.
- Các chỉ số của phương pháp longest matching thì đạt kết quả tuyệt đối. Việc này có thể giải thích như sau: Trong quá trình, nhóm đã tạo một thư viện đầy đủ các từ cần tách mà về căn bản, longest matching là phương pháp có độ chính xác cao dựa trên từ điển. Bộ dữ liệu của nhóm lại ít trường hợp từ ghép, nhiều từ đơn nên việc các chỉ số đánh giá đạt tuyệt đối do mang thiên hướng chủ quan nhiều. Còn trong các thư viện, việc tách từ sẽ được áp dụng phương pháp học máy và bộ huấn luyện lớn nên vẫn có thể sai sót trên bộ dữ liệu nhỏ như trong báo cáo này.

Vậy có thể nói trong mô hình này, phương pháp longest matching do nhóm xây dựng thủ công là hiệu quả nhất.

Tổng số từ gán nhãn đúng



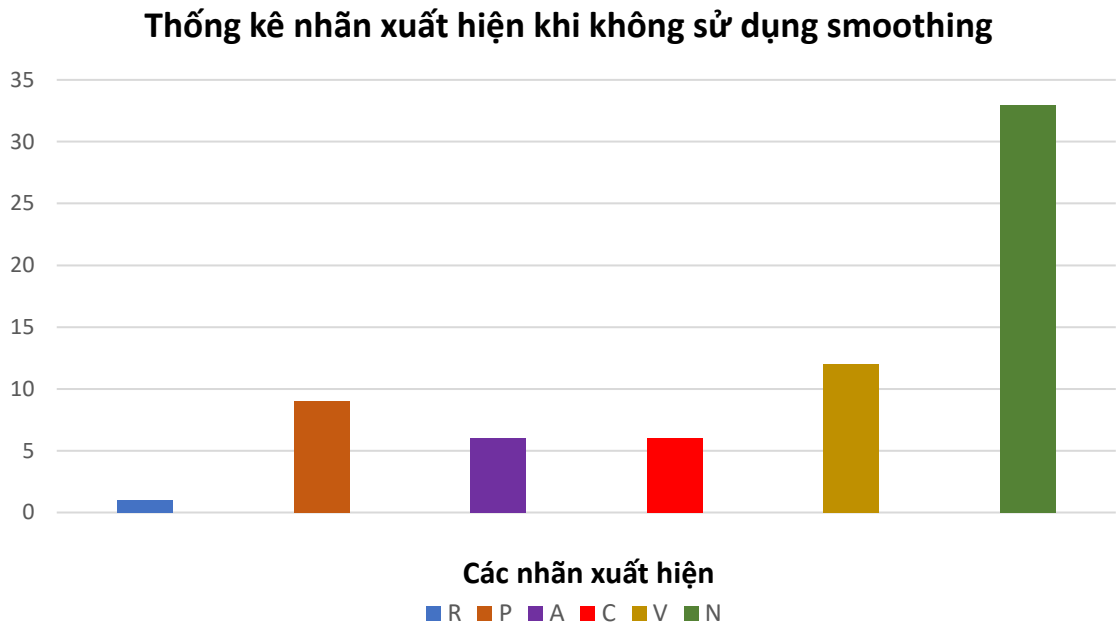
Thông qua biểu đồ "Số từ gán nhãn đúng", ta có thể nhận thấy rằng:

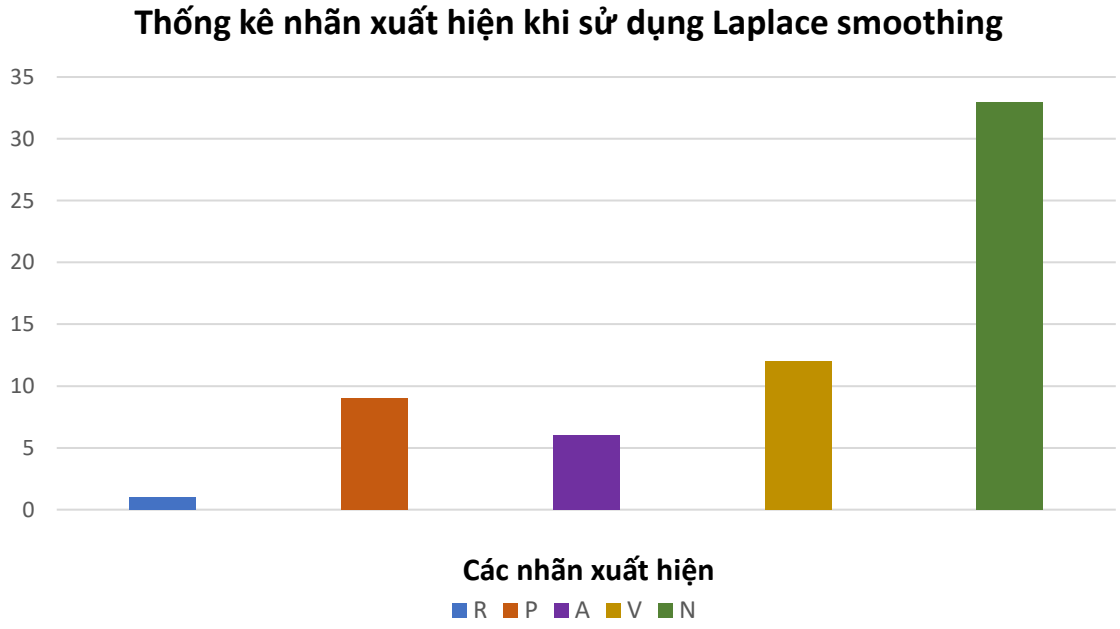
- Mô hình gán nhãn sử dụng HMM không smoothing trả về 60 kết quả đúng trên tổng là 67 của tập gold
- Mô hình gán nhãn sử dụng HMM có Laplace smoothing trả về 52 kết quả đúng trên tổng 67 nhãn của tập gold.

Vậy có thể thấy khách quan được là accuracy của HMM trong mô hình gán nhãn này khi không sử dụng smoothing sẽ cao hơn khi sử dụng Laplace smoothing. Điều này có thể giải thích như sau: Do trong Laplace smoothing, việc cộng 1 vào các nhãn và từ có thể đã tăng xác suất của các trường hợp không xảy ra trong mô hình. Vì vậy các chuỗi nhãn sai có thể có xác suất ban đầu là 0 nhưng lại được tăng lên và trong quá trình tính toán xác suất của các chuỗi nhãn ấy có thể cao hơn các chuỗi nhãn đúng. Việc này dẫn đến gán nhãn sai.

Vì accuracy là chỉ số duy nhất trong báo cáo này có thể dùng để đánh giá tổng quan toàn bộ mô hình gán nhãn nên nhóm cho rằng gán nhãn sử dụng Hidden Markov Model không sử dụng smoothing là hiệu quả nhất đối với chương trình cài đặt.

Các chỉ số đánh giá cho từng nhãn xuất hiện





Trường hợp	Nhãn xuất hiện					
	N	V	C	A	P	R
Không smoothing	33	12	6	6	9	1
Laplace smoothing	35	17	0	5	6	4

Bảng thống kê các nhãn xuất hiện trong trường hợp sử dụng HMM không smoothing và có Laplace smoothing

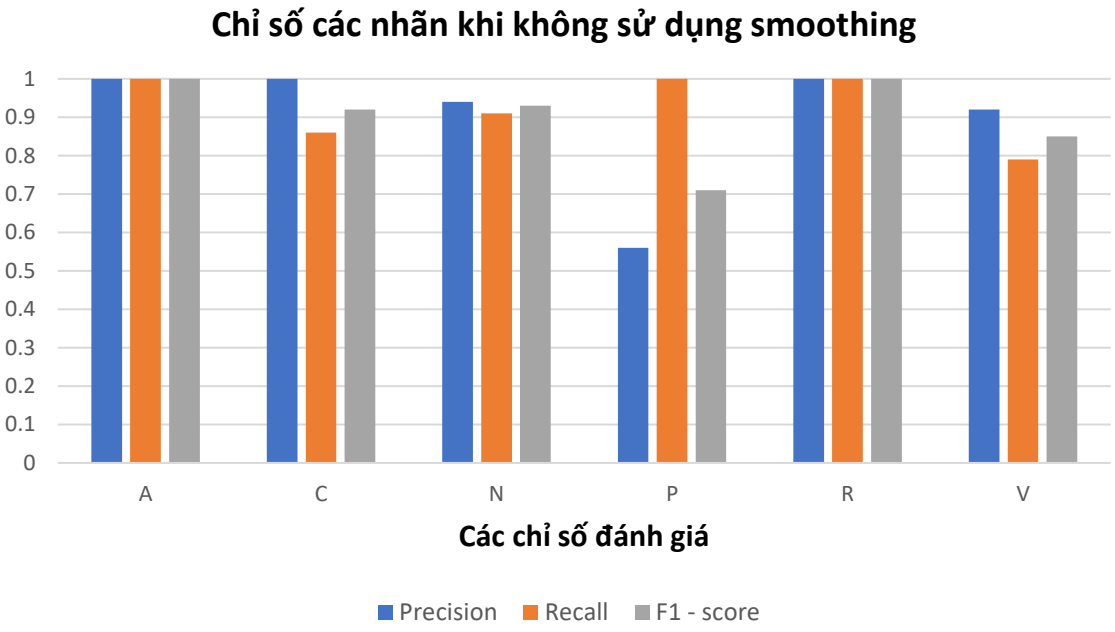
Thông qua các biểu đồ "Thống kê nhãn xuất hiện khi không sử dụng smoothing", "Thống kê nhãn xuất hiện khi sử dụng laplace smoothing", ta có thể nhận thấy rằng:

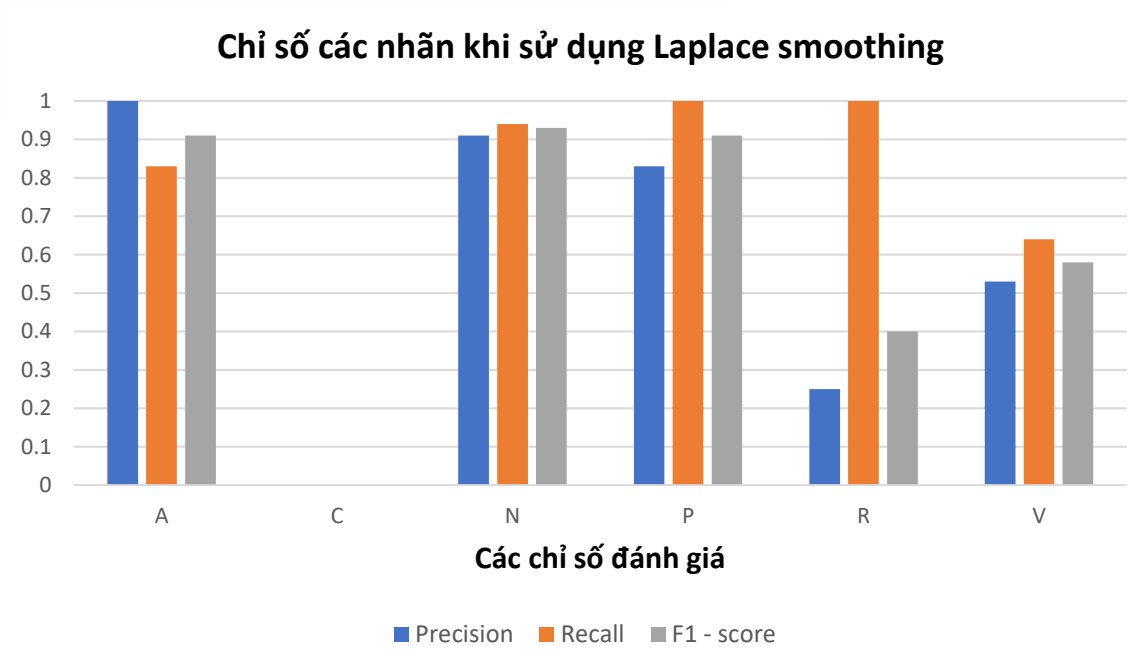
- Số nhãn N xuất hiện khi không sử dụng smoothing là 33, khi sử dụng Laplace smoothing là 35.
- Số nhãn V xuất hiện khi không sử dụng smoothing là 12, khi sử dụng Laplace smoothing là 17.
- Số nhãn C xuất hiện khi không sử dụng smoothing là 6, khi sử dụng Laplace smoothing là 0.
- Số nhãn A xuất hiện khi không sử dụng smoothing là 6, khi sử dụng Laplace smoothing là 5.
- Số nhãn P xuất hiện khi không sử dụng smoothing là 9, khi sử dụng Laplace smoothing là 6.
- Số nhãn R xuất hiện khi không sử dụng smoothing là 1, khi sử dụng Laplace smoothing là 4.

Đánh giá của nhóm:

- Trong mô hình gán nhãn HMM không sử dụng smoothing các nhãn có tần số xuất hiện từ thấp nhất đến cao nhất là: R – A – C – P – V – N.

- Trong mô hình gán nhãn HMM sử dụng Laplace smoothing các nhãn có tần số xuất hiện từ thấp nhất đến cao nhất là: R – A – P – V – N.
- Vậy N là nhãn xuất hiện nhiều nhất, R là nhãn xuất hiện ít nhất trong bộ thử nghiệm sau khi gán nhãn ở cả hai trường hợp không smoothing và có Laplace smoothing.
- Trong trường hợp có Laplace smoothing, nhãn C không xuất hiện so với bộ gold.
- Ngoài ra, mô hình đã không sử dụng hết số nhãn trong tập nhãn được trình bày ở chương 4. Cụ thể là các nhãn: ['E', 'I', 'L', 'M', 'S', 'T', 'X', 'F']





Thông qua các biểu đồ "Chỉ số các nhãn khi không sử dụng smoothing", "Chỉ số các nhãn khi sử dụng Laplace smoothing", ta có thể nhận thấy rằng:

- Trông mô hình HMM không sử dụng smoothing:
  - Nhãn A có chỉ số precision bằng 1, chỉ số recall bằng 1, chỉ số F1 – score bằng 1
  - Nhãn C có chỉ số precision bằng 1, chỉ số recall bằng 0.86, chỉ số F1 – score bằng 0.92
  - Nhãn N có chỉ số precision bằng 0.94, chỉ số recall bằng 0.91, chỉ số F1 – score bằng 0.93
  - Nhãn P có chỉ số precision bằng 0.56, chỉ số recall bằng 1, chỉ số F1 – score bằng 0.71
  - Nhãn R có chỉ số precision bằng 1, chỉ số recall bằng 1, chỉ số F1 – score bằng 1
  - Nhãn V có chỉ số precision bằng 0.92, chỉ số recall bằng 0.79, chỉ số F1 – score bằng 0.85

- Trông mô hình HMM sử dụng Laplace smoothing:
  - Nhãn A có chỉ số precision bằng 1, chỉ số recall bằng 0.83, chỉ số F1 – score bằng 0.91
  - Nhãn C có chỉ số precision bằng 0, chỉ số recall bằng 0, chỉ số F1 – score bằng 0
  - Nhãn N có chỉ số precision bằng 0.91, chỉ số recall bằng 0.94, chỉ số F1 – score bằng 0.93
  - Nhãn P có chỉ số precision bằng 0.83, chỉ số recall bằng 1, chỉ số F1 – score bằng 0.91
  - Nhãn R có chỉ số precision bằng 0.25, chỉ số recall bằng 1, chỉ số F1 – score bằng 0.4
  - Nhãn V có chỉ số precision bằng 0.53, chỉ số recall bằng 0.64, chỉ số F1 – score bằng 0.58

Từ các chỉ số trên, nhóm đã tổng kết lại một số nhận xét như sau:

- Trong mô hình HMM không sử dụng smoothing, nhìn tổng quan thì chỉ số precision và recall của các nhãn cao và không chênh lệch nhau nhiều. Trừ nhãn P, chỉ số recall của nhãn này cao trong khi precision chỉ ở mức trung bình. Do đó, chỉ số F1 – score của các nhãn cao.
- Trong mô hình HMM có sử dụng Laplace smoothing, sự chênh lệnh xảy ra khá nhiều ở các nhãn. Cụ thể, nhãn C có các chỉ số đều bằng 0 do không hề xuất hiện sau khi chạy thực nghiệm. Nhãn R thì lại chênh lệch precision và recall khá lớn. Nhãn V thì các chỉ số chỉ ở mức trung bình.

Việc tính toán các chỉ số này đã được trình bày cụ thể ở ví dụ chương 4 nên ở đây sẽ không nhắc lại. Do các chỉ số của các nhãn trong mô hình gán nhãn bằng HMM không sử dụng smoothing đồng đều hơn các chỉ số của các nhãn trong mô hình gán nhãn bằng HMM khi sử dụng Laplace smoothing, ta có thể nói việc không sử dụng smoothing sẽ cho ra kết quả các nhãn trong bộ thử nghiệm ít chênh lệch khi so với bộ gold hơn là khi áp dụng Laplace smoothing.

5. Nhận xét

Từ những nhận xét ở phần Kết quả thực nghiệm, nhóm xin tổng hợp lại các nhận xét sau:

- Về phần tách từ: Phương pháp longest matching trả về kết quả tuyệt đối nên là tốt nhất so với sử dụng thư viện.
- Về phần gán nhãn: mô hình HMM khi không sử dụng smoothing sẽ có độ chính xác cao hơn mô hình HMM sử dụng Laplace smoothing.
- Các nhãn của mô hình HMM không sử dụng smoothing sẽ đồng đều hơn các nhãn của mô hình HMM sử dụng Laplace smoothing.

Vậy mô hình phụ hợp nhất với đề tài của nhóm là:

Longest matching + HMM (không smoothing)

Chương 6. KẾT LUẬN

1. Nhận xét về mô hình

Phương pháp longest matching sẽ có độ chính xác phụ thuộc hoàn toàn vào từ điển nên việc liệt kê các từ xuất hiện trong bộ dữ liệu để tạo từ điển sẽ ảnh hưởng rất nhiều đến độ chính xác của mô hình. Việc sử dụng các phương pháp máy học có thể sẽ thuận tiện và linh hoạt hơn khi thay đổi và sửa chữa mô hình. Tuy vậy, độ chính xác cao là điểm nổi bật hơn cả của phương pháp này.

Mô hình Markov ẩn sẽ có độ chính xác phụ thuộc vào các xác suất chuyển trạng thái giữa các nhãn, xác suất của nhãn tương ứng với từ. Do đó, để việc gán nhãn có kết quả đúng nhất cần huấn luyện đa dạng các trường hợp cho mô hình. Các thuật toán có trong HMM rất tốn kém, cả về bộ nhớ lẫn thời gian tính toán. Tuy vậy, đây là một mô hình tương đối đơn giản, dễ hiểu và linh hoạt.

2. Bài học kinh nghiệm

Sau khi hoàn thành xong đề tài này, các thành viên trong nhóm chúng tôi đã đều có khả năng trả lời các câu hỏi liên quan tới mô hình tách từ sử dụng longest matching và gán nhãn sử dụng Hidden Markov Model. Đây cũng là một thách thức do là một đồ án chuyên ngành mà các thành viên trong nhóm lại chưa hoàn thành hết các kiến thức đại cương ở bậc Đại học. Bài báo cáo đi sát với toàn bộ nội dung thành viên đã thực hiện cũng như sưu tầm được những kiến thức cần phải phân tích rõ ràng. Trong quá trình nghiên cứu và tìm hiểu, chúng tôi đã thu được những kết quả thực nghiệm hữu ích thông qua việc áp dụng các phương pháp do nhóm đề xuất. Qua việc đánh giá các phương pháp, nhóm cũng đã thu được một số kinh nghiệm quý báu từ mô hình, tùy thuộc vào đặc điểm và yêu cầu của từng bài toán cụ thể mà lựa chọn phương pháp sao cho hợp lý. Mục đích của nhóm khi thực hiện thử nghiệm các trường hợp khác nhau khi cài đặt mô hình là để tìm ra phương pháp/mô hình tối ưu nhất với độ chính xác cao nhất có thể và đáp ứng được các yêu cầu đề ra. Ngoài ra, để có thể củng cố cho việc nghiên cứu, phát triển, chúng tôi sẽ cố gắng thu tập những thông tin cần thiết, tạo ra nhiều tính năng mới, đồng thời để khắc phục những hạn chế còn tồn tại trong đề tài này và nghiên cứu thêm những cái mới, hay để có thể hoàn thiện đề tài nghiên cứu về sau

Qua đó cũng có thể đánh giá rằng, đồ án môn học "Gán nhãn từ loại bằng mô hình Hidden Markov" do nhóm thực hiện không chỉ là một đồ án dừng lại ở mức độ chuyên ngành dành cho các thành viên trong nhóm, mà còn là cơ hội giúp các thành viên trong nhóm chúng tôi trao dồi thêm nhiều kiến thức và kỹ năng bổ ích, giúp mỗi thành viên trong nhóm nâng cao khả năng làm việc tập thể, nâng cao được tính tương tác giữa các thành viên, từ đó giúp bản thân mỗi thành viên trong nhóm qua quá trình học tập,làm việc nhóm rèn luyện thêm cho mình những kĩ năng cần thiết cho công việc sau này của bản thân.

TÀI LIỆU THAM KHẢO

[1] http://viet.jnlp.org/kien-thuc-co-ban-ve-xu-ly-ngon-ngu-tu-nhien/thuat-toan-tach-tu-tokenizer/thuat-toan-tach-tu

[2] https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/hidden-markov-model

[3] https://web.stanford.edu/~jurafsky/slp3/A.pdf

[4] https://www.youtube.com/watch?v=n25JjoixM3I&list=PLLssT5z_DsK8BdawOVCCaTCO99Ya58ryR&ab_channel=ArtificialIntelligence-AllinOne