**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙤🙧🟍🙥🙦



**GÁN NHÃN TỪ LOẠI TIẾNG VIỆT**

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn: | | |
| Th.S Nguyễn Trọng Chỉnh | | |
| Sinh viên thực hiện: | | |
| STT | Họ tên | MSSV |
| 1 | Đặng Hoàng Quân | 18520339 |
| 2 | Phạm Phú Phước | 18520131 |
| 3 | Dương Quốc Lộc | 18521006 |

**TP. HỒ CHÍ MINH – 1/2021**

# MỤC LỤC

[MỤC LỤC 2](#_Toc62658264)

[CHƯƠNG 1: GIỚI THIỆU 4](#_Toc62658265)

[CHƯƠNG 2: THU THẬP DỮ LIỆU 5](#_Toc62658266)

[2.1. Nguồn thu thập 5](#_Toc62658267)

[2.2. Thông tin cơ bản 5](#_Toc62658268)

[CHƯƠNG 3: TÁCH TỪ 6](#_Toc62658269)

[3.1. Giới thiệu Longest Matching 6](#_Toc62658270)

[3.2. Mã giả: 6](#_Toc62658271)

[3.2.1. Kết quả tách từ: 7](#_Toc62658272)

[CHƯƠNG 4: XÂY DỰNG BỘ DỮ LIỆU VÀ GÁN NHÃN 8](#_Toc62658273)

[4.1 Xây dựng bộ dữ liệu 8](#_Toc62658274)

[4.2. Cơ sở lý thuyết 9](#_Toc62658275)

[4.2.2. Thuật toán Hidden Markov Model 9](#_Toc62658276)

[4.2.3. Thuật toán Perceptron 12](#_Toc62658277)

[4.3. Tiền xử lí dữ liệu 13](#_Toc62658278)

[4.4. Thực nghiệm 13](#_Toc62658279)

[CHƯƠNG 5: KẾT QUẢ 14](#_Toc62658280)

[5.1. Các độ đo được sử dụng 14](#_Toc62658281)

[5.1.1. Độ đo Accuracy 14](#_Toc62658282)

[5.1.2. Độ đo Precision 14](#_Toc62658283)

[5.1.3. Độ đo Recall 14](#_Toc62658284)

[5.1.4. Độ đo F1-score 14](#_Toc62658285)

[5.2. Kết quả 15](#_Toc62658286)

[5.3. Nhận xét 15](#_Toc62658287)

[CHƯƠNG 6: KẾT LUẬN 16](#_Toc62658288)

# CHƯƠNG 1: GIỚI THIỆU

Part of speech (POS) tagging là một trong những phương pháp quan trọng của xử lý ngôn ngữ tự nhiên, cũng như trong việc hiểu nội dung câu hoặc văn bản. POS là thuật ngữ truyền thống để chỉ các loại từ được phân biệt về mặt ngữ pháp trong một ngôn ngữ.Trong quá trình phát triển chúng ta quen với việc xác định từ loại trong văn bản.Đọc một câu chúng ta có thể xác định rõ tử loại như là danh từ ,động từ hoặc tính từ… Để xác định từ rõ từ loại trong câu thường phức tạp hơn nhiều trong việc ánh xạ các từ qua từ điển. Đó là bởi vì một từ có thể được gán rất nhiều từ loại dựa vào ngữ cảnh của văn bản. Thật khó để chúng ta xác định một từ đó thuộc từ loại nào dựa vào một ngữ liệu nhất định vì tất cả ngữ cảnh mới và từ mới mỗi ngày liên tục xuất hiện đó cũng là vấn đề cho việc gán từ loại thủ công.

Trong đồ án này nhóm sẽ sử dung mô hình HiddenMarkov kết hợp thuật toán Viterbi để gán nhãn từ loại tiếng Việt và so sánh độ chính xác với VnCoreNLP

# CHƯƠNG 2: THU THẬP DỮ LIỆU

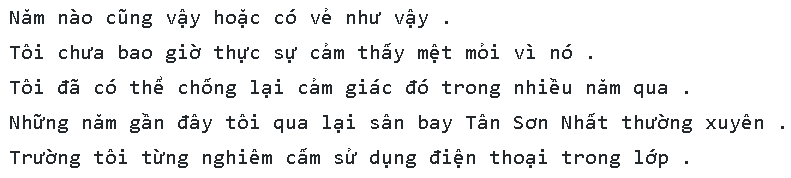
## Nguồn thu thập

Gồm các câu bất kỳ được thu thập từ Internet.

## Thông tin cơ bản

Bộ dữ liệu gốc với tên sentences.txt:

* Số lượng câu: 60 câu.
* Câu có số từ nhiều nhất: 41 từ.
* Câu có số từ ít nhất: 8 từ.



Hình 2: Một số câu có trong bộ dữ liệu gốc.

# CHƯƠNG 3: TÁCH TỪ

Bài toán tách từ là bài toán quan trọng đối với tiếng Việt. Khác với tiếng Anh, một từ tiếng Việt có thể được tạo bởi nhiều hơn một âm. Ví dụ từ (word) “cá\_nhân” được tạo lên bởi 2 âm (syllable) là “cá” và “nhân”. Trong khi hai từ đơn “cá” và từ đơn “nhân” lại có thể mang ý nghĩa khác.

Do vậy, tách từ tiếng Việt là bước quan trọng chúng ta cần thực hiện trước khi đưa dữ liệu vào các bước tiếp theo, ví dụ như word embedding.

## Các bước triển khai

Nhóm sẽ thực hiện tách từ bán thủ công: sử dụng thư viện VnCoreNLP để tách từ trước sau đó sẽ phân công các thành viên kiểm tra thủ công lại kết quả tách từ và sửa lại các từ bị sai để thu được bộ dữ liệu tách từ đúng nhất.

Sau đó, nhóm sẽ sử dụng bộ dữ liệu này và đánh giá kết quả tách từ của 2 phương pháp: sử dụng thuật toán Longest Matching và sử dụng thư viện VnCoreNLP.

## Thuật toán Longest Matching

### Giới thiệu

Longest Matching là thuật toán dựa trên tử tưởng tham lam. Nó xét các tiếng từ trái qua phải, các tiếng đầu tiên dài nhất có thể mà xuất hiện trong từ điển sẽ được tách ra làm một từ. Với vị trí âm tiết hiện tại sẽ kiểm tra xem từ đó và 2 âm tiếp theo có thể ghép thành 1 từ có nghĩa hay không bằng cách kiểm tra trong từ điển tri\_gram. Nếu không thể tạo ra được từ có nghĩa từ 3 âm tiết thì ta tiếp tục kiểm tra xem âm tiết hiện tại và âm tiếp theo có thể ghép được thành một từ có nghĩa hay không bằng cách kiểm tra trong từ điển bi\_gram. Thuật toán sẽ dừng khi xét hết các tiếng trong câu.

### Mã giả

current\_word \_id = 0 (Đặt từ hiện tại ở đầu câu):

While (curret\_word\_id+1) != len(text) {

xét từ hiện tại với 2 từ sau nó trong từ điển tri\_gram

check if {“word, word+1, word+2”} is in tri\_gram

if true: take {word, word1, word2; current\_word\_id += 3}🡪 3 từ đó tạo thành từ ghép

xét từ hiện tại với tiếng sau trong từ điển trong từ điển bi\_gram

else check if: take {“word, word+1”} is in bi\_gram

if true: take {word, word1; current\_word\_id += 2}🡪 2 tiếng đó tạo thành từ ghép

else => {take word; current\_word\_id += 1} 🡪 tiếng đó là từ đơn

}

### Ưu và nhược điểm

Ưu điểm thuật toán này chính là cài đặt đơn giản, độ phức tạp tính toán hợp lý, không yêu cầu dữ liệu huấn luyện. Tuy nhiên, phương pháp này sẽ khó có thể xử lý được các tình huống nhập nhằng nhất định như việc lặp từ trong câu cũng như xử lý tình huống xuất hiện từ mới không tồn tại trong từ điển

## Đánh giá kết quả tách từ

Bảng 1: Tổng hợp kết quả tách từ bằng phương pháp WFST.

|  |  |  |
| --- | --- | --- |
| Dữ liệu | Độ chính xác | Độ phủ |
| Từ câu 1 đến câu 10 | 0.9906 | 0.9952 |
| Từ câu 11 đến câu 20 | 0.9428 | 0.9705 |
| Từ câu 21 đến câu 30 | 1.0 | 1.0 |
| Từ câu 31 đến câu 40 | 1.0 | 1.0 |
| Từ câu 41 đến câu 49 | 1.0 | 1.0 |
| Toàn bộ dữ liệu (49 câu) | 0.9859 | 0.9929 |

Qua bảng 1, có thể thấy rằng kết quả tách từ chỉ tồn tại sai sót ở 20 câu dữ liệu đầu tiên. Từ bị tách sai nhiều nhất đó là từ “*số\_nguyên\_tố*” khi nó thường được tách thành “*số nguyên\_tố*”. Vì quá trình tách từ vẫn có sai sót, nhóm em quyết định sẽ lấy bộ dữ liệu GOLD có được khi thực hiện tách từ bằng tay để tiến hành đến bước gán nhãn từ loại.

# CHƯƠNG 4: TẠO NGỮ LIỆU VÀ GÁN NHÃN

## 4.1 Tạo ngữ liệu

Sau khi có kết quả tách từ, nhóm tiến hành gán nhãn bán thủ công trên bộ dữ liệu. Nhóm sẽ sử dụng thư viện VnCoreNLP để gán nhãn trước, sau đó sẽ phân công các thành viên kiểm tra thủ công lại kết quả gán nhãn và sửa lại các nhãn bị sai để thu được tập dữ liệu gold cho 60 câu đã thu thập.

Tiếp theo, nhóm sẽ sử dụng 50 câu để làm tập train và 10 câu để làm tập test. Sau đây là danh sách nhãn từ loại:

*Bảng 2: Danh sách nhãn từ loại*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Nhãn** | **Tên** | **Ví dụ** |
| 1 | N | Danh từ | tiếng, nước, thủ đô, nhân dân, đồ đạc, cây cối, chim muông |
| 2 | Np | Danh từ riêng | Nguyễn Du, Việt Nam, Hải Phòng, Trường Đại học Bách khoa Hà Nội, Mộc tinh, Hoả tinh, Phật, Đạo Phật |
| 3 | Nc | Danh từ chỉ loại | con, cái, đứa, bức |
| 4 | Nu | Danh từ đơn vị | mét, cân, giờ, nắm, nhúm, hào, xu, đồng |
| 5 | Ni | Danh từ ký hiệu | A1, A4, 60A, 60B, 20a, 20b, ABC, ABCD |
| 6 | V | Động từ | ngủ, ngồi, cười; đọc, viết, đá, đặt;  thích, yêu, ghét, giống, muốn |
| 7 | A | Tính từ | tốt, xấu, đẹp; cao, thấp, rộng |
| 8 | P | Đại từ | tôi, chúng tôi, hắn, nó, y, đại nhân, đại  ca, huynh, đệ |
| 9 | L | Định từ | mỗi, từng, mọi, cái; các, những, mấy |
| 10 | M | Số từ | một, mười, mười ba; dăm, vài, mươi;  nửa, rưỡi |
| 11 | R | Phó từ | đã, sẽ, đang, vừa, mới, từng, xong, rồi;  rất, hơi, khí, quá |
| 12 | E | Giới từ | trên, dưới, trong, ngoài; của, trừ,  ngoài, khỏi, ở |
| 13 | C | Liên từ | vì vậy, tuy nhiên, ngược lại |
| 14 | Cc | Liên từ đẳng lập | và, hoặc, với, cùng |
| 15 | I | Thán từ | ôi, chao, a ha |
| 16 | T | Trợ từ | à, a, á, ạ, ấy, chắc, chăng, cho, chứ |
| 17 | B | từ vay mượn | Internet, email, video, chat |
| 18 | Y | Từ viết tắt | OPEC, WTO, HIV |
| 19 | X | Các từ không thể phân loại |  |
| 20 | Z | Yếu tố cấu tạo từ | bất, vô, phi |
| 21 | CH | Nhãn dành cho các loại dấu | . ! ? , ; : |

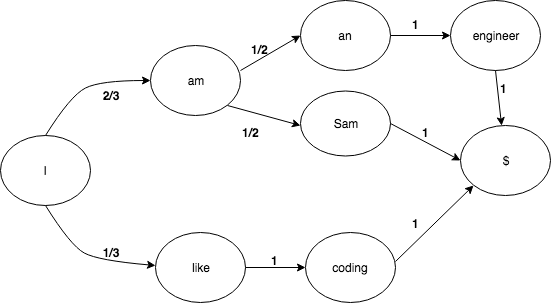
Quá trình gán nhãn do 2 thành viên của nhóm thực hiện độc lập với nhau. Với kích thước của bộ dữ liệu GOLD là 837 từ, quá trình gán nhãn cần hơn 3 giờ cho mỗi người.

## 4.2. Cơ sở lý thuyết

### 4.2.2. Mô hình Hidden Markov

Mô hình Markov ẩn (HMM) là một mô hình thống kê, lần đầu tiên được đề xuất bởi Baum L.E. (Baum và Petrie, 1966) và sử dụng quy trình Markov có chứa các tham số ẩn và không xác định. Trong mô hình này, các tham số rõ được dùng để xác định các tham số ẩn. Các thông số này sau đó được sử dụng để cho các phân tích sâu hơn. HMM là một loại của chuỗi Markov.

Markov Chain (Xích Markov), hay Visible Markov Model, là một dạng FSA được dùng để mô hình hóa xác suất của các biến ngẫu nhiên có quan hệ với nhau theo dạng chuỗi. Xích Markov có chứa 3 thành phần là tập các trạng thái, hàm chuyển đổi trạng thái có trọng số và trạng thái bắt đầu. Một ví dụ về Xích Markov như sau:



*Hình 9: Một ví dụ về Xích Markov.*

Từ 3 câu văn: ‘I am an engineer’, ‘I am Sam’, ‘I like coding’. Ta có thể vẽ ra đồ thị chuyển trạng thái như trên. Các giá trị xác suất được tính toán theo công thức

Trong đó:

* P là xác suất của việc chuyển từ trạng thái qi sang trạng thái qj.
* qi, qj là các trạng thái trong chuỗi.
* count(x) là số lần xuất hiện của x trong chuỗi.

Các thành phần của mô hình HMM:

* S = {s1, s2, .., sn} là tập các trạng thái ẩn
* Trạng thái đặc biệt: s0 là trạng thái bắt đầu.
* K = {k1, k2, .., km} là tập các giá trị quan sát
* A = {aij}, (i,j = 1..n) là ma trận chuyển trạng thái, trong đó aij là xác suất chuyển từ trạng thái si sang trạng thái sj.
* B = {bij} (i=1..n, j=1..m) là ma trận emisson (thể hiện), trong đó bij là xác suất trạng thái ẩn si thể hiện bằng giá trị quan sát kj.

Bậc của HMM: là số trạng thái ẩn trước đó được dùng để tính toán đến xác suất của trạng thái kế tiếp. Thông thường, số trạng thái ẩn được chọn là 1, 2, …

* Bậc 1: p(xT|x1..xT-1) = p(xT|xT-1)
* Bậc 2: p(xT|x1..xT-1) = p(xT|xT-2xT-1)

Từ dữ liệu đã gán nhãn, 2 ma trận chuyển trạng thái A và ma trận thể hiện B được tính toán dựa theo số lần xuất hiện của các nhãn và bậc của mô hình. Trong đó, xác xuất chuyển trạng thái với mô hình bậc k sẽ bằng:

Và xác xuất thể hiện của trạng thái ẩn là

Cần thực hiện smoothing Laplace + 1 để tránh các phép toán bằng 0 không mong muốn.

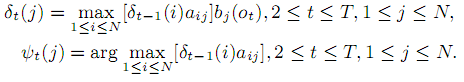
Từ 2 bảng ma trận A và B, ta có thể xác định chuỗi trạng thái ẩn X tương ứng với chuỗi trạng thái quan sát được O. Ta cần tìm sao cho p(X, O|M) là lớn nhất và dùng thuật toán Viterbi để thực hiện

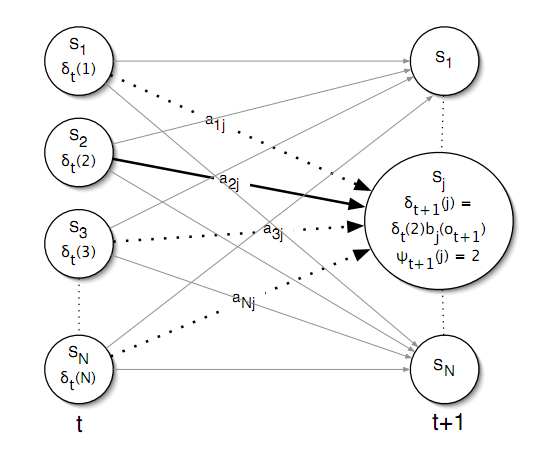
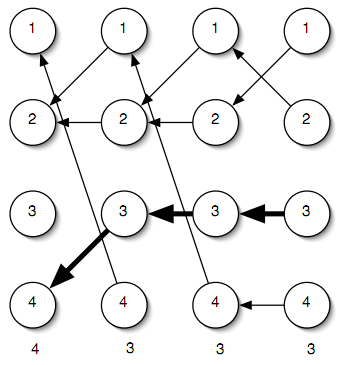
Thuật toán Viterbi gồm 4 bước như sau

* Bước khởi tạo:



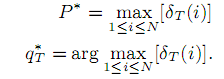
* Bước đệ quy:



*Hình 10: Bước đệ quy và bước Backtracking của thuật toán Viterbi.*

* Phần Termination:

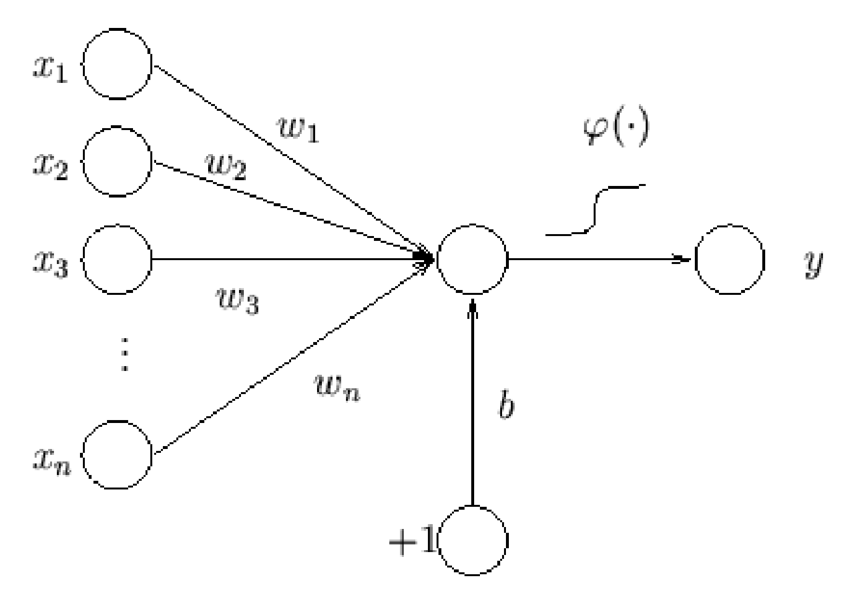


* Bước Backtracking backtracking:



### 4.2.3. Thuật toán Viterbi

Thuật toán Perceptron được phát minh vào năm 1958 tại Phòng thí nghiệm Hàng không Cornell bởi Frank Rosenblatt. Đây là một trong những thuật toán mạng neuron cơ bản và đơn giản nhất khi chỉ có 3 lớp là một lớp đầu vào, một lớp ẩn và một lớp đầu ra.



*Hình 11: Kiến trúc của thuật toán Perceptron.*

Thuật toán Perceptron gồm các bước sau: khởi tạo, tính toán giá trị và cập nhật tham số**.**

* Khởi tạo các giá trị trọng số. Thường khởi tạo bằng 0 hoặc các giá trị nhỏ.
* Tính toán giá trị đầu ra theo công thức
* Cập nhật tham số:

Trong đó w là trọng số, x là giá trị dữ liệu và là đạo hàm của W theo y.

Khi qua đến cuối lớp ẩn, ta tính được giá trị đầu ra, từ đó tính được hàm mất mát với công thức: . Trong đó Y là giá trị thực tế và y là giá trị dự đoán. Việc sử dụng vòng lặp để lặp đến khi lượng mất mát này giảm đến một mức chấp nhận được.

## 4.4. Thực nghiệm

Thông qua nhiều lần chạy mô hình để tìm ra các bộ tham số tối ưu nhất. Đối với thuật toán đầu tiên là Transformation Based Learning, các tập luật mà thông qua thực nghiệm mà nhóm em tìm được cho kết quả tốt nhất là:

*Bảng 3: Các luật được sử dụng trong TBL.*

|  |  |
| --- | --- |
| **STT** | **Dạng luật** |
| 1 | Template(Word([0])) |
| 2 | Template(Word([2])) |
| 3 | Template(Word([-1]), Pos([-1])) |
| 4 | Template(Word([-2]), Pos([-2])) |
| 5 | Template(Word([-1]), Pos([0])) |
| 6 | Template(Word([0]), Pos([1])) |
| 7 | Template(Word([-2]), Pos([0])) |
| 8 | Template(Word([1])) |

Đối với thuật toán Perceptron, các siêu tham số được sử dụng là learning\_rate = 0.001, hàm kích hoạt là hàm relu có công thức là max(0, x), số vòng lặp là 500 vòng.

# CHƯƠNG 5: KẾT QUẢ

## 5.1. Các độ đo được sử dụng

### 5.1.1. Độ đo Accuracy

Độ đo accuracy là tỉ lệ giữa số điểm dữ liệu được dự đoán đúng và tổng số điểm dữ liệu trong tập dữ liệu. Độ chính xác giúp ta đánh giá hiệu quả dự báo của mô hình trên một bộ dữ liệu. Độ chính xác càng cao thì mô hình của chúng ta càng chuẩn xác. Công thức chung: Accuracy = . Hệ số accuracy cho bảng kết quả trên là (55+900)/1000 = 90.5%.

*Bảng 4: Ví dụ về các độ đo.*

|  |  |  |  |
| --- | --- | --- | --- |
| Dự đoán/Thực tế | | Thực tế | |
| BAD (Positive) | GOOD (Negative) |
| Dự đoán | BAD(Positive) | 55 (True Positive - TP) | 50 (False Positive - FP) |
| GOOD (Negative) | 45 (False Negative - FN) | 850 (True Negative - TN) |
| Tổng cộng |  | 100 | 900 |

### 5.1.2. Độ đo Precision

Độ đo Precision cho biết có bao nhiêu phần trăm trong các trường hợp được dự báo là positive là đúng. Công thức cho độ đo này là Precision = . Như vậy, độ đo precision cho bảng kết quả trên là 55/(50+55) = 52.4%.

### 5.1.3. Độ đo Recall

Recall đo lường tỷ lệ dự báo chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm positive. Công thức cho độ đo này là Recall = .

Như vậy, độ đo recall cho bảng kết quả trên là 55/(55+45) = 55%.

### 5.1.4. Độ đo F1-score

F1 Score là trung bình điều hòa giữa precision và recall. Do đó nó đại diện hơn trong việc đánh g**iá** độ chính xác trên đồng thời precision và recall.

Công thức của F1-score là:F1 = .

Như vậy, F1 của bảng kết quả trên là 2\*

## 5.2. Kết quả

Sau khi chạy thực nghiệm, nhóm em đã được kết quả sau trên bộ dữ liệu gold.

*Bảng 5: Kết quả của 3 thuật toán*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuật toán** | **Accuracy** | **Precision** | **Recall** | **F1** |
| 1 | TBL | 0.8125 | 0.8095 | 0.7857 | 0.7809 |
| 2 | HMM | 0.9375 | 0.9523 | 0.9761 | 0.9584 |
| 3 | Perceptron | 0.9617 | 0.9242 | 0.9586 | 0.9245 |

## 5.3. Nhận xét

Thuật toán Transformation Based Learning cho kết quả thấp nhất giữa 3 thuật toán. Các độ đo chỉ ở mức 0.80 với Accuracy đạt 81.25%. Độ đo F1 đạt 78.09%. Đây là một kết quả đạt ở mức chấp nhận được nhưng kết quả này có thể cải thiện hơn nữa trong 2 thuật toán tiếp theo.

Ở thuật toán Hidden Markov Model, tỉ lệ chính xác của mô hình này đạt 93.75%, một con số khá cao. Nhưng 3 độ đo còn lại trên thuật toán này còn đạt cao hơn với Precision, Recall, F1 lần lượt là 95.23%, 97.61% và 95.84%. Như vậy, F1 của HMM là cao nhất giữa 3 thuật toán.

Mặc dù thuật toán Perceptron cho kết quả Accuracy cao nhất (96.17%) nhưng 3 độ đo còn lại trên thuật toán này lại không đạt tốt bằng thuật toán Hidden Markov Model. Độ đo recall đạt 95.86%, cao nhất giữa 3 độ đo nên F1 đạt 92.45%.

Một điều đặc biệt giữa các độ đo là Recall cho kết quả cao hơn Precision trên 2 thuật toán Hidden Markov Model và Perceptron. Điều đó cho thấy số lượng dự đoán đúng trên tổng số lượng thực tế là đúng là tốt hơn so với tỉ lệ trên độ đo Precision.

# CHƯƠNG 6: KẾT LUẬN

Qua đồ án môn học Xử lí Ngôn ngữ Tự nhiên này, nhóm em đã làm được nhiều phần trong bài toán gán nhãn từ loại cho Tiếng Việt. Từ bộ dữ liệu thu thập trên một bài báo ở trang báo điện tử Zingnews. Nhóm em đã thực hiện các bước xử lí như tách từ bằng WFST, gán nhãn thủ công qua công cụ VLSP.

Sau khi có được bộ dữ liệu hoàn chỉnh, nhóm em tiến hành thực nghiệm trên ba mô hình là Transformation Based Learning, Hidden Markov Model và Perceptron. Mô hình TBL cho kết quả thấp nhất giữa ba mô hình với Accuracy đạt 81.25%. Mô hình Hidden Markov Model dù cho độ đo Accuracy thấp hơn mô hình Perceptron nhưng cho F1, Recall và Perceptrong với F1 đạt 95.84%. Độ đo Accuracy cao nhất ở thuật toán Perceptron với kết quả 96.17%.

Hạn chế của đồ án này là bộ dữ liệu không thật sự lớn. Dữ liệu là một phần rất quan trọng, không thể thiếu trong các dự án về máy học. Chất lượng dữ liệu ảnh hưởng lớn đến chất lượng của mô hình nhưng vì trong phạm vi đồ án này nên nhóm em chỉ có thể xây dựng một bộ dữ liệu nhỏ để phục vụ cho bài toán. Khi phải giải quyết với dữ liệu chưa có trong bộ dữ liệu thì độ chính xác của mô hình sẽ không cao như bảng kết quả trên. Vì vậy, hướng phát triển của bài toán này có thể là mở rộng dữ liệu hơn nữa, xây dựng một bộ dữ liệu lớn hơn, bao quát đầy đủ trường hợp ngữ cảnh trong tiếng Việt.