TRƯỜNG ĐẠI HỌC NHA TRANG

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**TÌM HIỂU TÌM HIỂU TRÍCH TRỌN ĐẶC TRƯNG**

**TÚI ĐỰNG TỪ (BAG OF WORDS)**

Giảng viên: TS. Đinh Đồng Lưỡng

Nghiên cứu sinh: Cao Viết Thắng

Lớp: 61CHCNTT

Khánh Hòa, tháng 08/2020

# MỤC LỤC

Đề mục Trang

Trang bìa i

[MỤC LỤC 2](#_Toc49589330)

[CHƯƠNG 1. GIỚI THIỆU 3](#_Toc49589331)

[CHƯƠNG 2. MÔ HÌNH CHUNG CHO CÁC BÀI TOÁN HỌC MÁY 4](#_Toc49589332)

[2.1. TRAINING PHASE 5](#_Toc49589333)

[2.1.1. Feature Extractor 5](#_Toc49589334)

[2.1.2. Main Algorithms 6](#_Toc49589335)

[2.2. TESTING PHASE 6](#_Toc49589336)

[CHƯƠNG 3. BAG-OF-WORDS 7](#_Toc49589337)

[3.1. Tìm hiểu về Bag of Words 7](#_Toc49589338)

[3.2. Xử lý Part-Of-Speech (POS) với thuật toán Conditional Random Fields 10](#_Toc49589339)

[3.2.1. Part Of Spech là gì? 10](#_Toc49589340)

[3.2.2. Các phương pháp gắn thẻ POS 10](#_Toc49589341)

[3.2.3. Trường xác suất có điều kiện (Conditional Random Fields – CRFs) 11](#_Toc49589342)

[3.3. Bag-of-Words trong Computer Vision 13](#_Toc49589343)

[CHƯƠNG 4. KẾT LUẬN 16](#_Toc49589344)

[TÀI LIỆU THAM KHẢO 17](#_Toc49589345)

# GIỚI THIỆU

Khi chúng ta học lý thuyết về Machine Learning, đa phần ta đều giả sử các điểm dữ liệu được biểu diễn bằng các vector, được gọi là *feature vector* hay *vector đặc trưng*, có độ dài bằng nhau, và cùng là vector cột hoặc vector hàng. Tuy nhiên, trong các bài toán thực tế, mọi chuyện không được tốt đẹp như vậy!

Với các bài toán về Computer Vision, các bức ảnh là các ma trận có kích thước khác nhau. Thậm chí để nhận dạng vật thể trong ảnh, ta cần thêm một bước nữa là *object detection*, tức là tìm cái khung chứa vật thể chúng ta cần dự đoán. Ví dụ, trong bài toán nhận dạng khuôn mặt, chúng ta cần tìm được vị trí các khuôn mặt trong ảnh và *crop* các khuôn mặt đó trước khi làm các bước tiếp theo. Ngay cả khi đã xác định được các khung chứa các khuôn mặt (và có thể resize các khung đó về cùng một kích thước), ta vẫn phải làm rất nhiều việc nữa vì hình ảnh của khuôn mặt còn phụ thưộc vào góc chụp, ánh sáng, … và rất nhiều yếu tố khác nữa.

Các bài toán NLP (Natural Language Processing - Xử lý ngôn ngữ tự nhiên) cũng có khó khăn tương tự khi độ dài của các văn bản là khác nhau, thậm chí có những từ rất hiếm gặp hoặc không có trong từ điển. Cũng có khi thêm một vài từ vào văn bản mà nội dung của văn bản không đổi hoặc hoàn toàn mang nghĩa ngược lại. Hoặc cùng là một câu nói nhưng tốc độ, âm giọng của mỗi người là khác nhau, thậm chí của cùng một người nhưng lúc ốm lúc khỏe cũng khác nhau.

Khi làm việc với các bài toán Machine Learning thực tế, nhìn chung chúng ta chỉ có được dữ liệu thô (raw) chưa qua chỉnh sửa, chọn lọc. Chúng ta cần phải tìm một phép biến đổi để loại ra những dữ liệu nhiễu (noise), và để đưa dữ liệu thô với số chiều khác nhau về cùng một chuẩn (cùng là các vector hoặc ma trận). Dữ liệu chuẩn mới này phải đảm bảo giữ được những thông tin đặc trưng (features) cho dữ liệu thô ban đầu. Không những thế, tùy vào từng bài toán, ta cần *thiết kế* những phép biến đổi để có những features phù hợp. Quá trình quan trọng này được gọi là *Feature Extraction*, hoặc *Feature Engineering*, một số tài liệu tiếng Việt gọi nó là *trích chọn đặc trưng*.

# MÔ HÌNH CHUNG CHO CÁC BÀI TOÁN HỌC MÁY

Phần lớn các bài toán Machine Learning có thể được thể hiện trong hình vẽ dưới đây:



Hình 1: Mô hình chung cho các bài toán Machine Learning.

Có hai phases lớn là Training phase và Testing phase.

**Với các bài toán Supervised learning, ta có các cặp dữ liệu (*input, output*), với các bài toán Unsupervised learing, ta chỉ có *input* mà thôi.**

## TRAINING PHASE

Có hai khối có nền màu xanh lục chúng ta cần phải thiết kế: Feature Extrctor và Main Algorythms

### Feature Extractor

**ĐẦU RA**

Ta sẽ đề cập đầu ra của khối này trước vì mục đích của Feature Engineering là tạo ra một Feature Extractor biến dữ liệu thô ban đầu thành dữ liệu phù hợp với từng mục đích khác nhau.

**ĐẦU VÀO**

* ***raw training input***. Raw input là tất cả các thông tin ta biết về dữ liệu.

Ví dụ: với ảnh thì là giá trị của từng pixel; với văn bản thì là từng từ, từng câu; với file âm thanh thì nó là một đoạn tín hiệu; với cơ sở dữ liệu **Iris** thì nó là độ dài các cánh hoa và đài hoa, … Dữ liệu thô này thường không ở dạng vector, không có số chiều như nhau. Thậm chí có thể có số chiều như nhau nhưng số chiều quá lớn, như một bức ảnh màu 1000 pixel x 1000 pixel thì số *elements* đã là 3×106*3×106* (3 vì ảnh màu thường có 3 channels: Red, Green, Blue). Đây là một con số quá lớn, không lợi cho lưu trữ và tính toán.

* **(optional) *output* của *training set***. Trong các bài toán Unsupervised learning, ta không biết *output* nên hiển nhiên sẽ không có đầu vào này. Trong các bài toán Supervised learning, có khi dữ liệu này cũng không được sử dụng.

Ví dụ: nếu *raw input* đã có cùng số chiều rồi nhưng số chiều quá lớn, ta muốn giảm số chiều của nó thì cách đơn giản nhất là *chiếu* vector đó xuống một không gian có số chiều nhỏ hơn bằng cách lấy một ma trận ngẫu nhiên nhân với nó. Ma trận này thường là ma trận *béo* (số hàng ít hơn số cột, tiếng Anh - fat matrices) để đảm bảo số chiều thu được nhỏ hơn số chiều ban đầu. Việc làm này mặc dù làm mất đi thông tin, trong nhiều trường hợp vẫn mang lại hiệu quả vì đã giảm được lượng tính toán ở phần sau. Đôi khi *ma trận chiếu* không phải là ngẫu nhiên mà có thể được *học* dựa trên toàn bộ *raw input*, ta sẽ có bài toán tìm ma trận chiếu để lượng thông tin mất đi là ít nhất. Trong nhiều trường hợp, dữ liệu *output* của *training set* cũng được sử dụng để tạo ra Feature Extractor. Ví dụ: trong bài toán classification, ta không quan tâm nhiều đến việc mất thông tin hay không, ta chỉ quan tâm đến việc những thông tin còn lại có đặc trưng cho từng class hay không. Ví dụ, dữ liệu thô là các hình vuông và hình tam giác có màu đỏ và xanh. Trong bài toán phân loại đa giác, các output là *tam giác* và *vuông*, thì ta không quan tâm tới màu sắc mà chỉ quan tâm tới số cạnh của đa giác. Ngược lại, trong bài toán phân loại màu, các class là *xanh* và *đỏ*, ta không quan tâm tới số cạnh mà chỉ quan tâm đến màu sắc thôi.

* **(optional) *Prior knowledge about data***: Đôi khi những giả thiết khác về dữ liệu cũng mang lại lợi ích. Ví dụ, trong bài toán classification, nếu ta biết dữ liệu là (gần như) ***linearly separable*** thì ta sẽ đi tìm một ma trận chiếu sao cho ở trong không gian mới, dữ liệu vẫn đảm bảo tính *linearly separable*, việc này thuận tiện hơn cho phần classification vì các thuật toán linear, nhìn chung, đơn giản hơn.

Sau khi *học* được feature extractor thì ta cũng sẽ thu được *extracted features* cho *raw input data*. Những *extracted features* này được dùng để huấn luyện các thuật toán Classification, Clustering, Regression,… ở phía sau.

### Main Algorithms

Khi có được *extracted features* rồi, chúng ta sử dụng những thông tin này cùng với (optional) *training output* và (optional) *prior knowledge* để tạo ra các mô hình phù hợp.

**Chú ý:** Trong một số thuật toán cao cấp hơn, việc *huấn luyện* feature extractor và main algorithm được thực hiện cùng lúc với nhau chứ không phải từng bước như trên.

**Một điểm rất quan trọng: khi xây dựng bộ *feature extractor* và *main algorithms*, chúng ta không được sử dụng bất kỳ thông tin nào trong tập *test data*. Ta phải giả sử rằng những thông tin trong *test data* chưa được nhìn thấy bao giờ. Nếu sử dụng thêm thông tin về *test data* thì rõ ràng ta đã *ăn gian*!**

## TESTING PHASE

Bước này đơn giản hơn nhiều. Với *raw input* mới, ta sử dụng feature extractor đã tạo được ở trên (tất nhiên không được sử dụng *output* của nó vì *output* là cái ta đang đi tìm) để tạo ra feature vector tương ứng. Feature vector được đưa vào *main algorithm* đã được học ở training phase để dự đoán *output*.

# BAG-OF-WORDS

## Tìm hiểu về Bag of Words

Khi làm việc với một văn bản, ta có những câu hỏi sẽ đặt ra:

* Với một văn bản thì feature vector sẽ có dạng như thế nào?
* Làm sao đưa các từ, các câu, đoạn văn ở dạng *text* trong các văn bản về một vector mà mỗi phần tử là một số?

Có một phương pháp rất phổ biến để trả lời những câu hỏi này. Phương pháp đó có tên là *Bag of Words (BoW)* (*Túi đựng Từ*).

Chúng ta bắt đầu bằng một ví dụ. Giả sử chúng ta có bài toán phân loại tin rác. Ta thấy rằng nếu một tin có chứa các từ *khuyến mại, giảm giá, trúng thưởng, miễn phí, quà tặng, tri ân, …* thì nhiều khả năng đó là một tin nhắn rác. Vậy phương pháp đơn giản nhất là *đếm* xem trong tin đó có bao nhiêu từ thuộc vào các từ trên, nếu nhiều hơn 1 ngưỡng nào đó thì ta quyết định đó là tin rác. (Tất nhiên bài toán thực tế phức tạp hơn nhiều khi các từ có thể được viết dưới dạng không dấu, hoặc bị cố tình viết sai chính tả, hoặc dùng ngôn ngữ teen). Với các loại văn bản khác nhau thì lượng từ liên quan tới từng chủ đề cũng khác nhau. Từ đó có thể dựa vào số lượng các từ trong từng loại để làm các vector đặc trưng cho từng văn bản.

Tôi xin lấy ví dụ cụ thể hơn về cách tạo ra vector đặc trưng cho mỗi văn bản dựa trên BoW và xin được lấy tiếng Anh làm ví dụ (nguồn **Bag of Words wiki**. Tiếng Việt khó hơn vì một từ có thể có nhiều âm tiết, tiếng Anh thì thường cứ gặp dấu cách là kết thúc một từ).

Giả sử chúng ta có hai văn bản đơn giản:

(1) John likes to watch movies. Mary likes movies too.

và

(2) John also likes to watch football games.

Dựa trên hai văn bản này, ta có danh sách các từ được sử dụng, được gọi là *từ điển* với 10 *từ* như sau:

["John", "likes", "to", "watch", "movies", "also", "football", "games", "Mary", "too"]

Với mỗi văn bản, ta sẽ tạo ra một vector đặc trưng có số chiều bằng 10, mỗi phần tử đại diện cho số từ tương ứng xuất hiện trong văn bản đó. Với hai văn bản trên, ta sẽ có hai vector đặc trưng là:

(1) [1, 2, 1, 1, 2, 0, 0, 0, 1, 1]

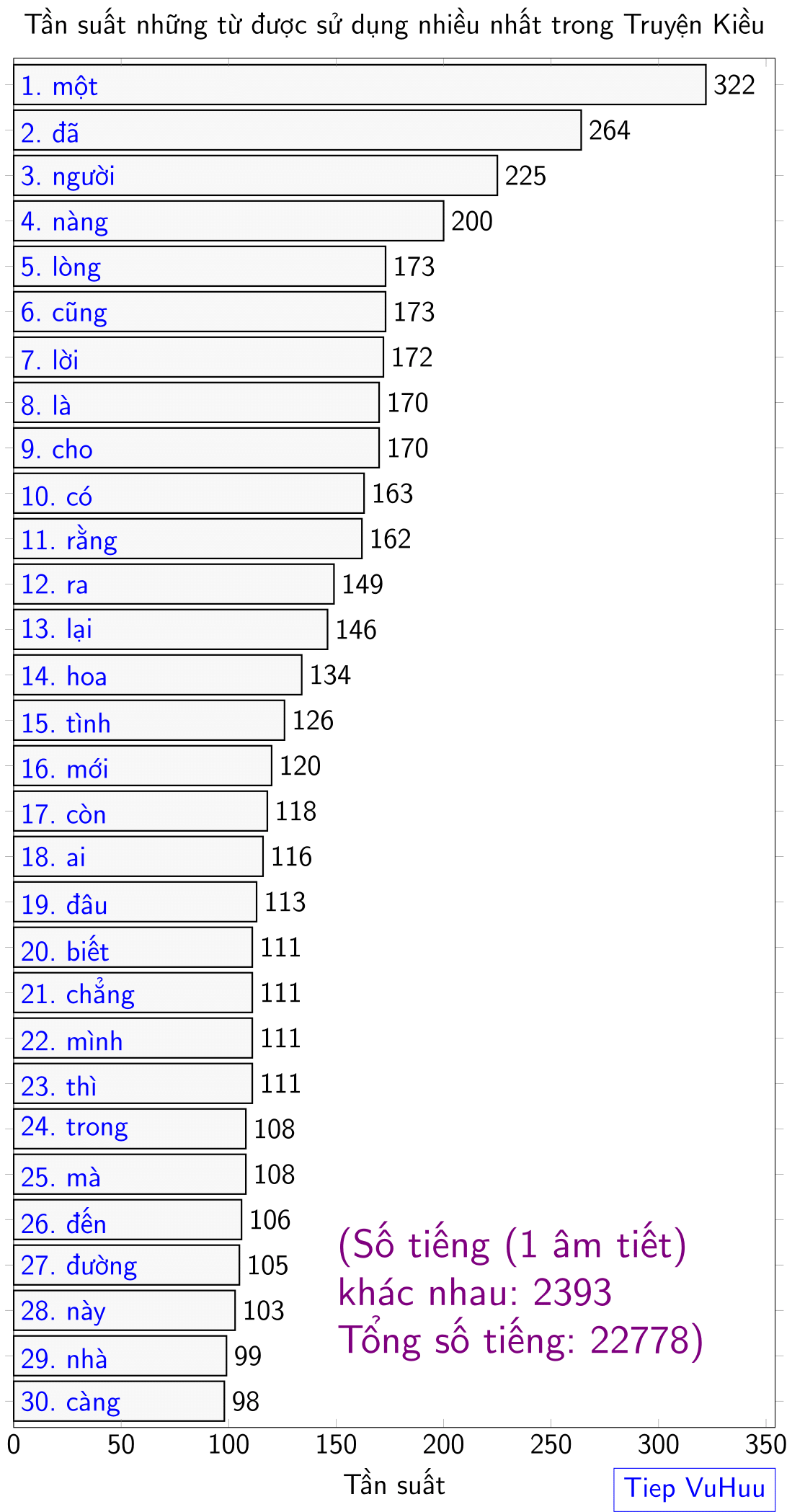
(2) [1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

Văn bản (1) có 1 từ “John”, 2 từ “likes”, 0 từ “also”, 0 từ “football”, … nên ta thu được vector tương ứng như trên.

Có một vài điều cần lưu ý trong BoW:

* Với những ứng dụng thực tế, *từ điền* có nhiều hơn 10 từ rất nhiều, có thể đến một trăm nghìn hoặc cả triệu, như vậy vector đặc trưng thu được sẽ rất *dài*. Một văn bản chỉ có 1 câu, và 1 tiểu thuyết nghìn trang đều được biểu diễn bằng các vector có số chiều bằng 100 nghìn hoặc 1 triệu.
* Có rất nhiều từ trong từ điển không xuất hiện trong một văn bản. Như vậy các vector đặc trưng thu được thường có rất nhiều phần tử bằng 0. Các vector có nhiều phần tử bằng 0 được gọi là *sparse vector* (sparse hiểu theo nghĩa là *thưa thớt, rải rác*, tôi xin phép chỉ sử dụng khái niệm này bằng tiếng Anh). Để việc lưu trữ được hiệu quả hơn, ta không lưu cả vector đó mà chỉ lưu *vị trí* của các phần tử khác 0 và *giá trị* tương ứng. *Lưu ý: nếu có hơn 50% số phần tử khác 0, việc làm này lại phản tác dụng!*
* Thi thoảng có những từ hiếm gặp không nằm trong từ điển, vậy ta sẽ làm gì? Một cách thường được dùng là *mở rộng* vector đặc trưng thêm 1 phần tử, gọi là phẩn tử *<Unknown>.* Mọi từ không có trong từ điền đều được coi là *<Unknown>.*
* Nghĩ kỹ một chút, những từ hiếm đôi khi lại mang những thông tin quan trọng nhất mà chỉ loại văn bản đó có. Đây là một nhược điểm của BoW. Có một phương pháp cải tiến khác giúp khắc phục nhược điểm này có tên là Term Frequency-Inverse Document Frequency (TF-IDF) dùng để xác định tầm quan trọng của một từ trong một văn bản dựa trên toàn bộ văn bản trong cơ sở dữ liệu (corpus).
* Nhược điểm lớn nhất của BoW là nó không mang thông tin về thứ tự của các từ. Cũng như sự liên kết giữa các câu, các đoạn văn trong văn bản. Ví dụ, ba câu sau đây: “*Em yêu anh không?*”, “*Em không yêu anh*”, và “*Không, (nhưng) anh yêu em*” khi được trích chọn đặc trưng bằng BoW sẽ cho ra ba vector giống hệt nhau, mặc dù ý nghĩa khác hẳn nhau.

**Ví dụ:** hình dưới đây là tần suất sử dụng các từ (coi mỗi âm tiết là một từ) trong Truyện Kiều nếu ta chỉ sử dụng 30 từ có tần suất cao nhất:



Hình 2: Bag of Words cho Truyện Kiều với 30 từ có tần suất cao nhất.

## Xử lý Part-Of-Speech (POS) với thuật toán Conditional Random Fields

### Part Of Spech là gì?

Ta đã biết, trong lĩnh vực Xử lý ngôn ngữ tự nhiên hầu hết các mô hình cơ bản đều được xây dựng dựa trên phương pháp **Bag of Words**. Nhưng, các mô hình như vậy không thể xác định được các mối quan hệ về cú pháp giữa các từ.

Ví dụ: với một mô hình phân tích cảm xúc được xây dựng trên phương pháp Bag of Words. Ta không thể xác định được sự khác biệt trong câu :”I like you”, trong đó, động từ “like” là một động từ chỉ tình cảm tích cực (tôi thích bạn). Và tương tự, với một câu khác: “I am like you” thì động từ “like” là là chỉ sự giống nhau giữa hai chủ thể (tôi giống bạn) chứ không chỉ tình cảm.

Vậy có cách nào ta có thể cải thiện thuật toán Bag of Words không?

Part of Speech (POS – gắn nhãn từ vựng) là một phương pháp trong xây dựng cây cú pháp mà tại đó ta có thể xác định NERs (các danh từ riêng/chung) và trích xuất quan hệ giữa các từ. Gắn thẻ POS cũng rất cần thiết để xác định được định dạng từ gốc (các dạng từ trong tiếng anh).

Gắn nhãn POS là xác định loại từ của một từ vựng trong ngữ cảnh đoạn văn đó. Để xác định được vấn đề này không hề đơn giản. Vì với một từ cụ thể, trong mỗi ngữ cảnh khác nhau sẽ có những ý nghĩa khác nhau.

Bạn có thể nhìn ví dụ sau: “Give me your answer” thì answer là một danh từ. Nhưng với câu “Answer the question” thì answer là một động từ.

Để máy tính có thể thực sự hiểu nghĩa của một câu và trích xuất được cấu trúc ngữ pháp của nó, thì POS là một bước quan trọng.

### Các phương pháp gắn thẻ POS

Hiện nay có rất nhiều kĩ thuật gắn thẻ POS khác nhau:

* Lexical Based Methods : gắn nhãn POS mỗi từ theo dạng từ xuất hiện có tần suất cao nhất trong bộ dữ liệu.
* [Rule-Based](https://trituenhantao.io/tu-dien-thuat-ngu/rule-based/) Methods : gắn nhã POS dựa trên một quy tắc xác định. Ví dụ: trong tiếng anh, những từ có kết thúc bằng “ed” hoặc “ing” thường được gán là một động từ. Phương pháp [Rule-Based](https://trituenhantao.io/tu-dien-thuat-ngu/rule-based/) Methods có thể kết hợp với phương pháp Lexical Based Methods để gắn nhãn những từ có trong bộ train nhưng không có trong bộ test.
* Probabilistic Methods: Phương pháp dự theo xác suất. Phương pháp này gắn nhã POS dựa trên xác xuất xảy ra của một chuỗi nhãn cụ thể. Thuật toán Conditional Random Fields (CRFs) và Hidden Markov Models (HMMs) là hai phương pháp phổ biến nhất.
* Deep Learning Methods: Sử dụng mạng neural để gắn nhãn POS.

Trong phạm vi bài báo cáo này, ta sẽ xem xét đến thuật toán **Conditional Random Fields** (CRFs).

### Trường xác suất có điều kiện (Conditional Random Fields – CRFs)

CRFs là thuật toán xác suất có điều kiện. Sự khác biệt ở mô hình này là sự phân phối xác suất theo điều kiện P(y/x), và xác suất này sẽ cố gắng đưa về xác suất bình thường: P(y, x).

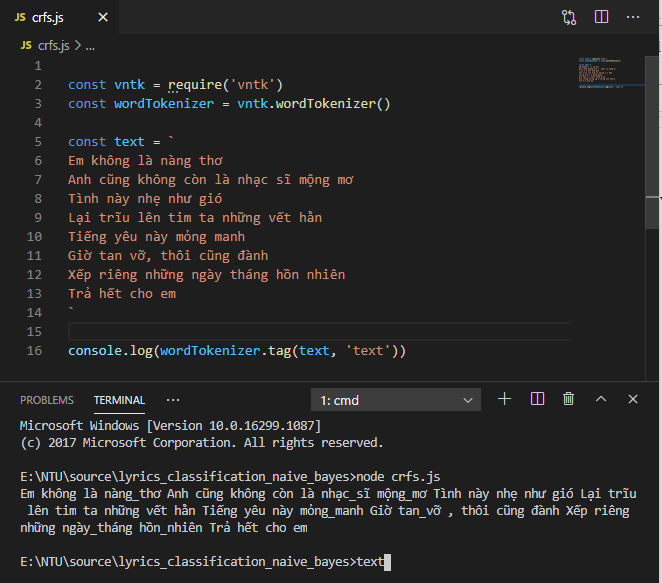
Mô hình hồi quy Logistic, SVM, CRFs là những thuật toán phân loại có điều kiện. Naive Bayes, HMMs là những thuật toán phân loại sinh mẫu. CRFs được sử dung cho việc gắn nhãn NREs và POS.

Trong thuật toán CRFs, đầu vào là tập hợp các thuộc tính (dạng số thực) từ tập dữ liệu đầu vào theo một quy tắc. Trọng số của biểu thức với các thuộc tính đầu vào cùng các nhãn đã được găn thẻ trước đó và task sẽ được dùng để dự đoán cho việc nhãn gắn hiện tại. Ta sẽ ước lượng trọng số sao cho chỉ số *likelihood* của nhãn trog bộ dữ liệu train là cực đại.

Hàm mục tiêu trong thuật toán sẽ xác định nhãn cho mỗi từ trong câu. Hãy tưởng tượng với một ví dụ sau: chữ cái đầu tiên của từ có phải là chữ cái viết hoa không?; các tiền tố và hậu tố của từ có dạng như thế nào?; …

Trong CRFs, chúng ta cũng xây dựng dự đoán nhãn từ hiện tại theo nhãn của các từ trước đó. Các trọng số của mô hình sẽ được hợp lý nhất.

Trong NodeJS, để áp dụng thuật toán Conditional Random Fields cho một đoạn văn bản tiếng việt ta sử dụng module vntk.



Hình 3. Kết quả xử lý POS sử dụng thuật toán CRFs bằng module vntk

## Bag-of-Words trong Computer Vision

BOW là một trong những phương pháp trích trọn đặc trưng trong học máy.

Bags of Words cũng được áp dụng trong Computer Vision với cách định nghĩa *words* và từ điển khác.

Xét các ví dụ sau:

**Ví dụ 1:**

Có hai class ảnh, một class là ảnh các khu rừng, một class là ảnh các sa mạc. Phân loại một bức ảnh là rừng hay sa mạc (giả sử ta biết rằng nó thuộc một trong hai loại này) một cách trực quan nhất là dựa vào màu sắc. Màu xanh nhiều thì là rừng, màu đỏ và vàng nhiều thì là sa mạc. Vậy chúng ta có thể có một mô hình đơn giản để trích chọn đặc trưng như sau:

* Với một bức ảnh, chuẩn bị một vector x*x* có số chiều bằng 3, đại diện cho 3 màu xanh (x1*x1*), đỏ (x2*x2*), và vàng (x3*x3*).
* Với mỗi điểm ảnh trong bức ảnh đó, xem nó gần với màu xanh, đỏ hay vàng nhất dựa trên giá trị của pixel đó. Nếu nó gần điểm xanh nhất, tăng x1*x1* lên 1; gần đỏ nhất, tăng x2*x2* lên 1; gần vàng nhất, tăng x3*x3* lên 1.
* Sau khi xem xét tất cả các điểm ảnh, dù cho bức ảnh có kích thước thế nào, ta vẫn thu được một vector có độ dài bằng 3, mỗi phần tử thể hiện việc có bao nhiêu pixel trong bức ảnh có màu tương ứng. Vector cuối này còn được gọi là vector histogram của bức ảnh tương ứng với ba màu xanh, đỏ, vàng. Dựa vào vector này, ta có thể quyết định bức ảnh đó là ảnh rừng hay sa mạc.

**Ví dụ 2:**

Trên thực tế, các bài toán xử lý ảnh không đơn giản như ví dụ 1 trên đây. Mắt người thực ra nhạy với các đường nét, hình dáng hơn là màu sắc. Một cái (ảnh) cây dù không có màu vẫn là một cái (ảnh) cây! Vì vậy, xem xét giá trị từng điểm ảnh một không mang lại kết quả khả quan vì lượng thông tin bị mất quá nhiều.

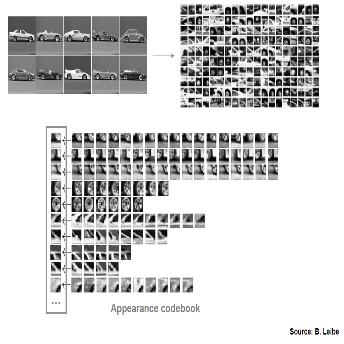
Có một cách khắc phục là thay vì xem xét một điểm ảnh, ta xem xét một *cửa sổ* nhỏ trong ảnh (trong Computer Vision, cửa sổ này được gọi là patch) là một hình chữ nhật chứa nhiều điểm ảnh gần nhau. Cửa sổ này đủ lớn để có thể chứa được các bộ phận có thể mô tả được vật thể trong ảnh.

Ví dụ với mặt người, các patch nên đủ lớn để chứa được các phần của khuôn mặt như mắt, mũi, miệng như hình dưới đây.



Hình 4: Bag of Words cho ảnh chứa mặt người.

Tương tự thế, với ảnh là ô tô, các patch thu được có thể là bánh xe, khung xe, cửa xe, … như hàng trên trong hình dưới đây.



Hình 5: Bag of Words cho ảnh ô tô.

Có một câu hỏi đặt ra là, trong xử lý văn bản, hai từ được coi là như nhau nếu nó được biểu diễn bởi các ký tự giống nhau. Vậy trong xử lý ảnh, hai patchés được coi là như nhau khi nào? Khi mọi pixel trong hai patches có giá trị bằng nhau sao?

Câu trả lời là không. Xác suất để hai patches giống hệt nhau từng pixel là rất thấp vì có thể một phần của vật thể trong một patch bị lệch đi vài pixel so với phần đó trong patch kia; hoặc phần vật thể trong patch bị méo, hoặc có độ sáng khác nhau, mặc dù ta vẫn nhìn thấy hai patches đó *rất giống nhau*. Vậy thì hai patch được coi là như nhau khi nào? Và *từ điển* ở đây được định nghĩa như thế nào?

Câu trả lời ngắn: hai patches là gần giống nhau nếu khoảng cách Euclid giữa hai vector tạo bởi hai patches đó gần nhau. Từ điển (codebook) sẽ có số phần tử do ta tự chọn. Số phần tử càng cao thì độ sai lệch càng ít, nhưng sẽ nặng về tính toán hơn.

Câu trả lời dài: chúng ta có thể áp dụng **K-means clustering**. Với rất nhiều patches thu được, giả sử ta muốn xây dựng một *codebook* với chỉ khoảng 1000 *words*. Vậy thì ta cho k=1000*k=1000* rồi thực hiện K-means clustering trên toàn bộ số patches thu được (từ tập training). Sau khi thực hiện K-means clustering, ta thu được 1000 clusters và 1000 centers tương ứng. Mỗi centers này được coi là một *words*, và tất cả những điểm rơi vào cùng một cluster được coi là cùng một bag. Với ảnh trong tập test data, ta cũng lấy các patches rồi xem chúng rơi vào những bags nào. Từ đó suy ra vector đặc trưng cho mỗi bức ảnh. Chú ý rằng với k=1000*k=1000*, mỗi bức ảnh sẽ được *mô tả* bởi một vector có số chiều 1000, tức là mỗi điểm dữ liệu bây giờ đã có số chiều bằng nhau, mặc dù ảnh thô đầu vào có thể có kích thước khác nhau.

# KẾT LUẬN

Feature Extraction là một bước quan trọng trong học máy, bag of words là một trong nhưng phương pháp trích trọn đặc trưng phổ biến của bài toán liên quan tới xử lý ngôn ngữ tự nhiên.

# TÀI LIỆU THAM KHẢO

**Tiếng Việt:**

* Website:

1. Vũ Hữu Tiệp, Bài 11: Giới thiệu về Feature Engineering, <https://machinelearningcoban.com/general/2017/02/06/featureengineering/>
2. Blog Trí tuệ nhân tạo, [NLP] Xử lý POS với thuật toán Conditional Random Fields,

<https://trituenhantao.io/kien-thuc/nlp-xu-ly-pos-voi-thuat-toan-conditional-random-fields/>

**Tiếng Anh:**

* Website:

1. Wikipedia, Bag-of-words Model,

<https://en.wikipedia.org/wiki/Bag-of-words_model>

1. Wikipedia, Feature Extraction,

<https://en.wikipedia.org/wiki/Feature_extraction>

1. Wikipedia, Conditional random field,

<https://en.wikipedia.org/wiki/Conditional_random_field>