**CÔNG TY CÔNG NGHỆ VÀ KỸ THUẬT STECH**

Logo

Description automatically generated

**ĐỒ ÁN SỐ 3:**

**“NLP – XỬ LÝ NGÔN NGỮ TỰ NHIÊN”**

**Thực tập sinh - Nguyễn Văn Đàn**

[**Dan.nguyen060201@stechvn.org**](mailto:Dan.nguyen060201@stechvn.org)

**Language Modeling - Mô hình ngôn ngữ và bài toán thêm dấu câu trong Tiếng Việt**

Mô hình ngôn ngữ là một thuật ngữ mà bất cứ ai đã, đang và sẽ tìm hiểu về Xử lí ngôn ngữ tự nhiên (NLP) đều biết và cần phải biết để có thể hiểu rõ hơn về cách thức mà một ngôn ngữ được xây dựng từ một bộ từ vựng, về cách đánh giá, cách xử lí đối với ngôn ngữ tự nhiên cũng như là tiền đề để đi sâu vào tìm hiểu các lĩnh vực sâu xa hơn như : **sửa lỗi chính tả, dịch máy, gán nhãn từ loại**, ... .

Một ứng dụng phổ biến của mô hình ngôn ngữ mà mọi người hầu như tiếp xúc nhiều nhất đó là việc **tự động gợi ý từ tiếp theo** trên thanh tìm kiếm của Google.

## Trong bài viết này, mình sẽ cùng các bạn tìm hiểu một cách chi tiết, rõ ràng nhất về bản chất của một mô hình ngôn ngữ cũng như survey qua các hướng xây dựng mô hình ngôn ngữ đang phổ biến hiện nay. Cuối bài viết, chúng ta cũng sẽ thử xây dựng một mô hình ngôn ngữ nho nhỏ trong tiếng việt và ứng dụng nó vào bài toán thêm dấu câu cho tiếng việt

## Định nghĩa

Mô hình ngôn ngữ là tập hợp các kiến thức trước đó về một ngôn ngữ nhất định, các kiến thức này có thể là các kiến thức về từ vựng, về ngữ pháp, về tần suất xuất hiện của các cụm từ, ... Một mô hình ngôn ngữ có thể được xây dựng theo hướng chuyên gia hoặc hướng dữ liệu.

Đó là định nghĩa chung nhất về mô hình ngôn ngữ, tuy nhiên khá mơ hồ và chung chung nhỉ. Do đó, mình sẽ nêu lại định nghĩa về mô hình ngôn ngữ dưới góc nhìn hẹp hơn: Về tần suất xuất hiện của cụm từ.

Mô hình ngôn ngữ là một phân bố xác suất trên các tập văn bản, cung cấp các thông tin về phân bố xác suất tiền nghiệm (prior distribution) p(w\_1...w\_n)*p*(*w*1​...*wn*​) với (w\_1, ..., w\_n )(*w*1​,...,*wn*​) là các từ vựng trong bộ từ điển của một ngôn ngữ nhất định. Nói đơn giản, mô hình ngôn ngữ có thể cho biết xác suất một câu (hoặc cụm từ) thuộc một ngôn ngữ là bao nhiêu.

Ví dụ: khi áp dụng mô hình ngôn ngữ cho tiếng Việt:

1. P[“hôm nay là thứ bảy”] = 0.001
2. P[“bảy thứ hôm là nay”] = 0

Ví dụ trên cho thấy, câu "hôm nay là thứ bảy" có khả năng là một câu tiếng Việt cao hơn so với câu "bảy thứ hôm là nay". Đây là ví dụ đơn giản nhất để các bạn có thể hình dung được một mô hình ngôn ngữ làm gì. Vậy câu hỏi tiếp theo được đặt ra là chúng ta tính cái xác suất p(w\_1...w\_n)*p*(*w*1​...*wn*​) kia như thế nào, hay làm thể nào để xây dựng mô hình ngôn ngữ ?

**Các hướng tiếp cận**

Để xây dựng mô hình ngôn ngữ, chúng ta có thể tiếp cận theo 3 hướng chính: Knowledge-based Language Model, Statistical Language Model (Count-based), và Neural Network Language Model (Continuous-space). Ngoài ra còn một số mô hình ngôn ngữ khác chẳng hạn như KenLM, ...

### Knowledge-based Language Model

Đây là những mô hình ngôn ngữ được xây dựng dựa trên những kiến thức đã được con người (cụ thể là các chuyên gia về ngôn ngữ học) tích luỹ, phân tích từ cú pháp một câu, cách chia động từ hoặc sự phân rã của tính từ ... .

Khi các kiến thức này được định nghĩa bằng các luật, nó sẽ xây dựng lên một mô hình ngôn ngữ, do đó Knowledge-based Language Model còn được gọi là rule-based language model.

* Grammatical - Ungrammatical
* Intra-grammatical - Extra-grammatical
* Non-grammatical - Out-of-grammatical
* Qualitative LM - Quantitative LM

Một số thách thức và bất lợi khi cố gắng xây dựng một Knowledge-based Language Model có thể kể đến như: (trích từ bài viết [Language Model là chi rứa?](https://viblo.asia/p/language-model-la-chi-rua-maGK7Vkb5j2))

* Khó xây dựng: Trước tiên, bạn cần phải là một chuyên gia ngôn ngữ học. Tiếp theo, bạn cần đủ khả năng để tổng hợp và tái cấu trúc lại các kiến thức của bạn về một ngôn ngữ thành tập các luật biểu diễn nó. (Thế thôi, chắc nghỉ game nhỉ )
* Chỉ nhận diện được các từ thuộc intra-grammatical: Intra-grammatical có thể hiểu như là văn viết (formal), trái ngược với extra-grammatical là văn nói (informal). Nếu học ngôn ngữ thì ngôn ngữ nào cũng có sự khác biệt giữa văn nói và văn viết.
* Thiếu tính tần số (Lack of frequencies): Độ phổ biến của câu từ đóng vai trò không kém quan trọng, chẳng hạn như "How to recognize speech" và "How to wreck a nice beach" đều đúng ngữ pháp nhưng câu đầu đúng hơn vì phổ biến hơn và câu sau có nghĩa hơi kì lạ.
* Chỉ phân biệt được hợp lý hay không: Do cấu trúc của mô hình nên kết quả của mô hình này đối với một câu là có hợp lý (hay đúng ngữ pháp) của một ngôn ngữ hay không chứ không có dự đoán hay gợi ý được từ.

Do một loạt các lí do trên, các hướng tiếp cận mình giới thiệu tiếp theo thường được quan tâm và tập trung phát triển hơn

### Statistical Language Model

Xây dựng mô hình ngôn ngữ dựa trên thống kê là việc cố gắng đi xác định giá trị của p(w\_1...w\_n)*p*(*w*1​...*wn*​) từ tập dữ liệu thu thập được.

p(w\_1...w\_n)*p*(*w*1​...*wn*​) hiểu đơn giản là xác suất cụm từ w\_1...w\_n*w*1​...*wn*​ thuộc một ngôn ngữ cụ thể, mà trong trường hợp với tập dữ liệu thu thập được, ta có thể tính

p(w\_1...w\_n) = \frac{count(w\_1...w\_n)}{N}*p*(*w*1​...*wn*​)=*Ncount*(*w*1​...*wn*​)​

với N là số lượng cụm từ có độ dài N trong tập dữ liệu. Do đó, Statistical Language Model còn được gọi là Count based Model

Hoặc trong trường hợp việc xác định N là khó thực hiện, chúng ta hoàn toàn có thể sử dụng công thức xác suất có điều kiện trong trường hợp này :

p(w\_1...w\_n) = p(w\_1)\*p(w\_2|w\_1)\*p(w\_3|w\_1w\_2)\*...\*p(w\_n|w\_1...w\_{n-1})*p*(*w*1​...*wn*​)=*p*(*w*1​)∗*p*(*w*2​∣*w*1​)∗*p*(*w*3​∣*w*1​*w*2​)∗...∗*p*(*wn*​∣*w*1​...*wn*−1​)

Trong đó.

p(w\_k|w\_1...w\_{k-1}) = \frac{count(w\_1...w\_k)}{count(w\_1...w\_{k-1})}*p*(*wk*​∣*w*1​...*wk*−1​)=*count*(*w*1​...*wk*−1​)*count*(*w*1​...*wk*​)​

Dựa theo cách thức xác định giá trị xác suất p(w\_1...w\_n)*p*(*w*1​...*wn*​), trong Statistical Language Model lại được chia thành 2 hướng tiếp cận nhỏ hơn

**N-gram Language Models**

Việc tính giá trị p(w\_1...w\_n)*p*(*w*1​...*wn*​) trong trường hợp n vô hạn, thực tế là vô cùng khó khăn. Để giảm độ phức tạp cho việc tính toán cũng như tạo ra một hướng đi khả thi để có thể mô hình hóa ngôn ngữ, mô hình n-gram ra đời. Mô hình n-gram giả định việc mô hình ngôn ngữ là một chuỗi Markov, thỏa mãn tính chất Markov. Chúng ta có tính chất Markov được định nghĩa như sau:

Một quá trình mang tính ngẫu nhiên có thuộc tính Markov nếu phân bố xác suất có điều kiện của các trạng thái tương lai của quá trình, khi biết trạng thái hiện tại, phụ thuộc chỉ vào trạng thái hiện tại đó

Hiểu một cách đơn giản, nếu giả định mô hình ngôn ngữ có thuộc tính Markov, ta có thể tính xác suất p(w\_1...w\_n)*p*(*w*1​...*wn*​) theo công thức sau :

p(w\_1...w\_n) = p(w\_1)\*p(w\_2|w\_1)\*p(w\_3|w\_1w\_2)\*...\*p(w\_n|w\_1...w\_{n-1}) = p(w\_1)\*p(w\_2|w\_1)\*p(w\_3|w\_2)\*...\*p(w\_n|w\_{n-1})*p*(*w*1​...*wn*​)=*p*(*w*1​)∗*p*(*w*2​∣*w*1​)∗*p*(*w*3​∣*w*1​*w*2​)∗...∗*p*(*wn*​∣*w*1​...*wn*−1​)=*p*(*w*1​)∗*p*(*w*2​∣*w*1​)∗*p*(*w*3​∣*w*2​)∗...∗*p*(*wn*​∣*wn*−1​)

Công thức trên là dạng giả định đơn giản nhất : mô hình ngôn ngữ là một mô hình markov bậc 1. Tuy nhiên, thực tế điều giả định là quá đơn giản, và các mô hình markov bậc 2, 3, 4 thường được sử dụng nhiều hơn. Điều giả định này đã giúp việc tính toán trở nên đơn giản hơn rất nhiều

Ví dụ, chúng ta có câu : W = "Today is Sarturday and towmorrow is Sunday". Khi đó với mô hình markov bậc 2

p(W) = \text{p(today)}\*...\*\text{p(sunday|today is sarturday and towmorrow is )} = \text{p(today)}\*...\*\text{p(sunday|towmorrow is )}*p*(*W*)=p(today)∗...∗p(sunday|today is sarturday and towmorrow is )=p(today)∗...∗p(sunday|towmorrow is )

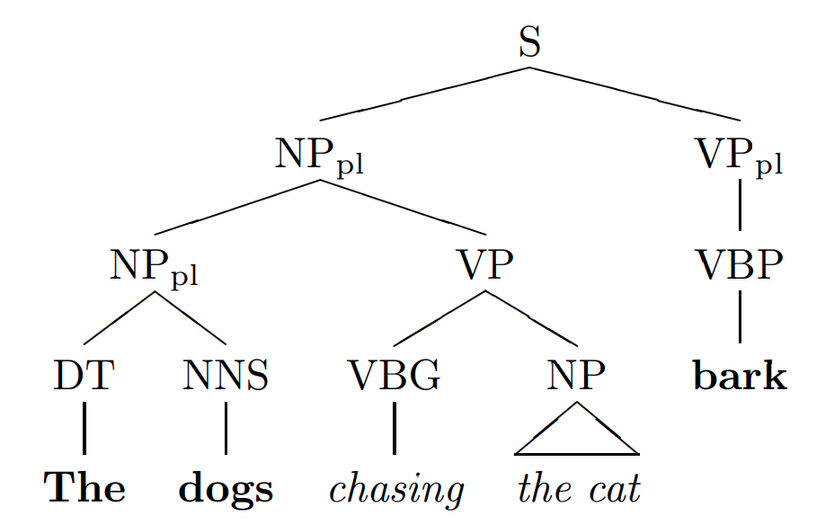
Dễ hình dung hơn rồi nhỉ, chúng ta sẽ chỉ xét các từ đứng gần từ đang xét thôi, còn lại thì bỏ qua =))

Khi giả định một mô hình ngôn ngữ là mô hình Markov bậc n-1, ta gọi các mô hình đó là mô hình n-gram. Việc tính toán cụ thể hơn, mình sẽ tiếp tục trình bày ở ngay bên dưới để các bạn có thể nắm rõ hơn 

**Structured Language Models**

Một nhược điểm của mô hình n-gram là không đánh giá được ngữ cảnh của toàn bộ câu, do đó, trong nhiều trường hợp, nó không thể đưa ra một xác suất chính xác. Một ví dụ có thể kể đến như W = "The dogs chasing the cat bark" (tạm dịch là Những con chó vừa nãy đuổi con mèo thì đang sủa). Động từ "bark" (sủa) ở đây nhận chủ ngữ là "The dogs", tuy nhiên, nếu sử dụng 3-gram, mô hình sẽ đi tính xác suất p(bark|the cat)*p*(*bark*∣*thecat*), và kết quả thu được sẽ là bằng 0.

Để giải quyết vấn đề này, Structured Language Models ra đời. Structured Language Models hay mô hình ngôn ngữ có cấu trúc, cố gắng hướng đến một hệ thống phân cấp cho các từ vựng trong một câu văn. Việc này kết hợp với n-gram sẽ cho kết quả bao quát được ngữ cảnh của toàn câu văn, cải thiện độ chính xác của mô hình



### Neural Network Language Model

Neural Network Language Model là những phương pháp mới nhất dựa trên mạng Neural để xây dựng mô hình ngôn ngữ, các mô hình này còn có một tên gọi khác là Continuous-space language models. Neural Network Language Model cũng được chia thành 2 hướng tiếp cận chính

**Feed-forward neural network based LM**: Feed-forward neural network được đề xuất để giải quyết 1 vấn đề khác của N-gram liên quan đến độ thưa thớt dữ liệu (khi tồn tại nhiều câu, cụm từ trong thực tế nhưng lại không được thu thập trong dữ liệu train). Mạng neural được sử dụng ở đây là 1 mạng neural 3 lớp, nhằm mục tiêu học được các tham số cần thiết để tính xác suất p(w\_n|w\_1...w\_{n-1})*p*(*wn*​∣*w*1​...*wn*−1​).

**Recurrent Neural Network Based LM**: Recurrent Neural Network lại tập trung giải quyết một khía cạnh khác về ngữ cảnh của toàn bộ câu, giúp giải quyết được các giới hạn về ngữ cảnh. Hiện nay, các mô hình ngôn ngữ dựa trên RNN hoặc các phát triển từ nó như LSTM, ... đang đạt được các kết quả State of the art.

Nếu như với mô hình Feed-forward neural network, dữ liệu đầu vào cần yêu cầu số từ cố định, tuy nhiên, điều này là không thể xảy ra trong thực tế, do các câu có độ dài ngắn khác nhau, Recurrent Neural Network cũng đồng thời khắc phục được nhược điểm này với việc chấp nhận đầu vào có độ dài bất kì.

# 2. Ứng dụng mô hình ngôn ngữ trong bài toán thêm dấu tiếng việt

OK, tạm xong phần survey khiến các bạn nhức não (và khiến mình nhức tay - do viết nhiều quá @@), trong phần này, chúng ta sẽ bắt tay xây dựng một mô hình ngôn ngữ cho Tiếng Việt để cảm thấy chút hứng thú với chủ đề này.

Note: Do trên Viblo hiện đã có một bài viết khá hay và chi tiết của tác giả Phạm Hoàng Anh về việc xây dựng mô hình ngôn ngữ cho tiếng Việt bằng mạng LSTM (tức là theo hướng Neural Network Language Model) : [Tạo Language Model để tự động sinh văn bản tiếng Việt](https://viblo.asia/p/tao-language-model-de-tu-dong-sinh-van-ban-tieng-viet-E375zxEjZGW), thế nên mình sẽ tập trung giới thiệu cụ thể hơn về 1 phương pháp khác : **N-gram model**

## Dữ liệu

Để tiện so sánh với phương pháp LSTM được xây dựng của tác giả Phạm Hoàng Anh, chúng ta sẽ sử dụng lại luôn bộ dữ liệu trong bài viết của tác giả. Tất nhiên, cũng sẽ hơi khác một xíu. Nếu tác giả chỉ sử dụng 1/4 bộ dữ liệu đấy thì chúng ta sẽ sử dụng full tất cả (train n-gram nhanh mà ^^).

Mình xin trích lại link bộ dữ liệu tại đây luôn <https://github.com/hoanganhpham1006/Vietnamese_Language_Model/blob/master/Train_Full.zip>

Giới thiệu sơ qua một chút :

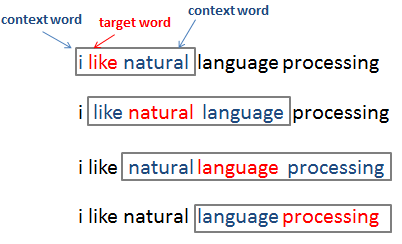
* Tập dữ liệu bao gồm 42744 bài viết, được thu thập từ các trang tin tức, báo chí, xã hội
* Gồm 8 thư mục về 8 chủ đề khác nhau: Chính trị xã hội, Đời sống, Kinh doanh, Pháp luật, Sức khỏe, Thế giới, thể thao, văn hóa
* Định dạng file .txt
* Ví dụ về 1 file trong thư mục Thế giới:

## Ý tưởng

Ý tưởng để chúng ta thực hiện bài toán thêm dấu Tiếng Việt bao gồm các bước sau :

* Xây dựng mô hình Trigram để đưa ra xác suất xuất hiện 1 từ nếu biết 2 từ đứng trước nó
* Đối với mỗi từ Tiếng Việt không dấu, tiến hành generate tất cả các trường hợp có thể điền dấu cho từ
* Sử dụng Greedy Search hoặc Beam Search để lấy ra được câu cho xác suất lớn nhất
* Test thử kết quả

OK, bắt tay vào công đoạn đầu tiên, là cơ sở cho sự thành bại của toàn bộ bài toán nào : Xây dựng mô hình Trigram.



## Xây dựng mô hình ngôn ngữ

* 1. Việc trước tiên, chúng ta cần load dữ liệu trước đã
  2. Tiếp theo là các bước tiền xử lí cơ bản để lấy đầu vào cho mô hình n-gram:
* Tách đoạn văn ra thành từng câu nhỏ
* Loại bỏ các dấu câu trong câu
* Tách từ (tokenize) với mỗi câu

Ở đây, mình chỉ có 1 lưu ý nhỏ đến vấn đề tokenize (tách từ). Có thể các bạn sẽ thắc mắc tại sao mình lại dùng word\_tokenize() của nltk mà không phải là word\_tokenize của underthesea hay ViTokenizer của pyvi, vốn là những tokenize thực sự tốt trong tiếng Việt. Câu trả lời rất đơn giản, là vì mô hình này mình xây dựng cho bài toán thêm dấu câu

Các bạn có thể hình dung vấn đề như sau :

* Với 1 câu bình thường : "Ứng dụng trong bài toán thêm dấu tiếng Việt"
* ViTokenizer hoặc underthesea.word\_tokenize : [ứng \_dụng, trong, bài\_toán, thêm, dấu , tiếng \_việt]
* Với 1 câu không dấu : "Ung dung trong bai toan them dau tieng viet"
* ViTokenizer hoặc underthesea.word\_tokenize : [ung, dung, trong, bai, toan, them, dau, tieng, viet]

# 3. Kết luận

Trong bài viết này, mình đã giới thiệu với các bạn về lí thuyết của mô hình ngôn ngữ cũng như survey qua các phương pháp xây dựng một mô hình ngôn ngữ hiện nay. Việc hiểu và ứng dụng được mô hình ngôn ngữ sẽ tạo ra cho các bạn một tấm nền vững chắc để bước đi trên con đường dài tìm hiểu về NLP. Về bài thực hành với bài toán thêm dấu câu trong Tiếng Việt, các bạn có thể tiếp tục cải thiện mô hình cả vè tốc độ và kết quả bằng nhiều cách : sử dụng một bộ dữ liệu đa dạng hơn, sử dụng phương pháp search tối ưu hơn, .... Các bạn cũng hoàn toàn có thể phát triển nó thành 1 ứng dụng web như trên trang <https://vietnameseaccent.com/> (vài dòng lệnh Flask là đủ rồi )