**Chuẩn hóa văn bản tiếng Việt**

**1. Định nghĩa bài toán**

Trong tổng hợp tiếng nói, cần đưa văn bản về dạng có thể đọc được. Văn bản trong thực tế bao gồm nhiều thành phần đặc biệt: số, ngày tháng, các từ viết tắt, tiền tệ, chuỗi chữ số, từ viết tắt, từ nước ngoài, số la mã, URL, email....

=> Được gọi chung là***NSWs***- not standard words

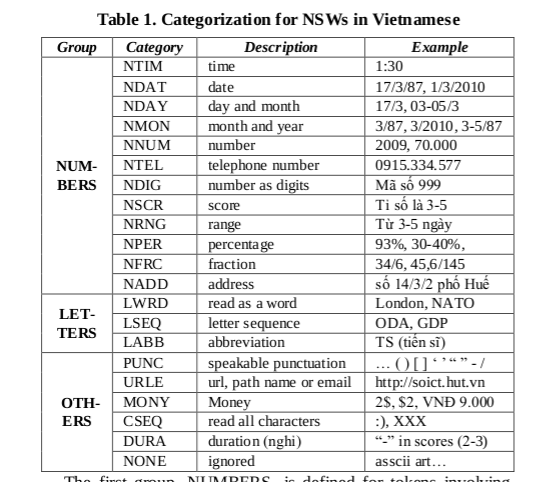
Các NSWs này không thể tìm ra cách phát âm theo luật “letter to sound”.

=> Vì vậy cần chuẩn hóa, đưa các NSWs về dạng đọc được của tiếng Việt.

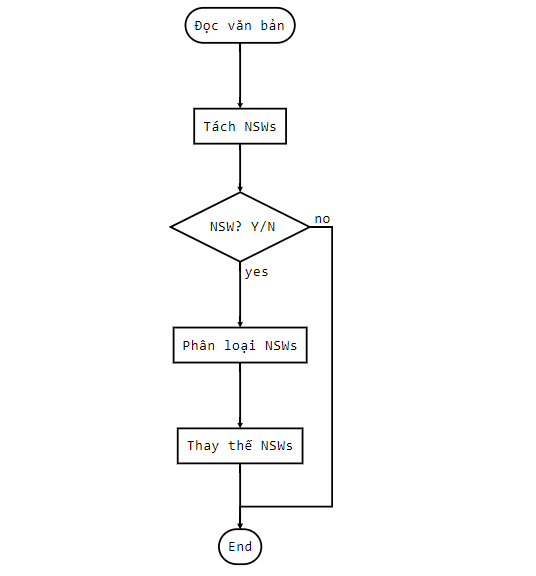
Ví dụ: “ngày 29/5” => “ngày hai mươi chín tháng năm”

PHUƠNG PHÁP: phân loại và chuẩn hóa từng loại

**2. Các loại NSWs**



**3. Mô hình chuẩn hóa**



***3.1. Tách NSWs:***

*3.1.1. Tách các token*

*­*- Tách các từ theo khoảng trắng

- Sửa các lỗi về khoảng trắng

Ví dụ: “1. 000.000” => “1.000.000”

“2 / 3” => “2/3”

- Sửa các lỗi về dấu câu

Ví dụ: “...đã xong.Viện CNTT ...”

”.. HN(Số 36) ..

- Khi phân tách các punc, chú ý giữ lại các đường link về URL

- Sửa cách các dạng viết tắt đơn vị tiền tệ về cuối của NSW

- Chỉnh sửa các âm tiết đặt sau dấu

Ví dụ: “oà” => “òa”

“oẹ” => “ọe”

*3.1.2. Phân loại các NSWs phức tạp*

- NSWs phức tạp là NSWs không thể phân loại vào các nhóm ở bảng 1, tuy nhiên có thể tách ra để phân loại vào các nhốm ở trong đó.

- Thường thì các NSWs phức tạp sẽ bao gồm cả chữ cái với chữ số (vd: “1m65”), chữ cái với dấu - hoặc dấu / (vd: “km/h”), chữ in hoa và chữ thường, chữ và số(“15.6-inch”).

=> dùng regrex để bắt và tách

- Các loại khác có thể là số, số điện thoại, ngày tháng, khoảng, tỷ số, ... và có thể được phân chia bằng các dấu ./- . (ví dụ: “16,17/12”)

=> tách từng token riêng biệt

***3.2. Phân loại NSWs***

- Các token ban đầu sẽ được chia ra thành 3 group lớn bao gồm: NUMBERS, LETTERS, OTHERS, được gán thuộc tính nsw, ví dụ nsw = “NUMBERS”. Ở mỗi nhóm chính token lại được phân loại thành từng loại trong bảng 1, và các NSW được cập nhật thành loại vừa được phân.

*3.2.1. Phân loại NUMBER*

- Trên thực mỗi token có thể phân thành nhiều loại, ví dụ “3-5” có thể là “ba đến năm” hoặc “ba tháng năm”, .... Vì vậy việc phân loại tùy thuộc vào ngữ cảnh.

=> Do đó, sử dụng cây quyết định để phân loại dựa và ngữ cảnh và thuộc tính định dạng. Ngữ cảnh sử dụng là 2 token trước và 2 token đứng sau

- Tuy nhiên, có một số dạng đặc biệt có thể phân loại luôn mà không cần thông qua ngữ cảnh (ví dụ: các token chứa % có thể phân loại vào NPER hoặc dạng như “15/12/2019-15/12/2020” có thể phân loại vào NDAY)

*3.2.2. Phân loại các từ dạng LETTER*

- Trên thực tế, thường khi viết tắt thì người viết sẽ viết hoa do đó những từ được viết hoa tất cả các kí tự thì ta có thể phân loại vào dạng LABB nếu có trong từ điển còn không thì ta để vào dạng LSEQ.

- Còn với các từ còn lại sẽ được phân vào LWRD để có thể diễn giải cách đọc hợp lý

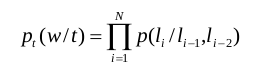
- Ngoài ra có thể sử dụng phương pháp thống kê để phân loại:

Tính xác suất mỗi loại label t cho token w:



Trong đó t là nhãn, thuộc 1 trong 3 loại LWRD, LESQ, LABB.

pt(w|t) được tính theo xác suất của tri-gram:



n = (l1,l2,....,ln) là xâu quan sát được tạo bởi n kí tự.

- Do quá trình test, những từ viết tắt nhập nhằng là không nhiều nên chưa cần dùng đến phương pháp phân loại đặc biệt.

*3.2.3. Phân loại nhóm OTHER*

*­*­- Do mỗi loại đều có định dạng riêng biệt nên có thể phân loại chính xác theo từng đặc trung:

+ Phân loại các punc thành 2 dạng là DURA và PUNC tùy theo từng loại punc.

+ Phân loại URLE chứa các cá @, https, www,…

+ Phân loại MONY với các token chứ kí hiệu và từ viết tắt của dạng tiền tệ.

+ Còn lại các dạng CSEQ, NONE thì sẽ được phân loại tùy theo trường hợp.

***4. Diễn giải các NSW***

***­***- Sinh ra các từ ngữ đầy đủ cho NSWs đã được phân loại. Mỗi NSW sẽ được thêm thuộc tính full để biểu diễn dạng full text của nó.

*4.1. NSWs dạng NUMBERS*

- Các dạng đã được phân loại có thể chứ thêm - để biểu diễn khoảng cách trong đó, khi đó chúng ta tách ra theo ‘-’ để diễn giải các NSW được chính xác và thay ‘-’ thành ‘đến’

Ví dụ: “30-60%” => NPER

- Diễn giải:

* NTIM: giờ, phút, giây (1:30 => một giờ ba mươi phút)
* NDAT: tháng, năm (18/01/2020 => mười tám tháng mười hai năm hai nghìn không trăm hai mươi)
* NDAY: tháng (tương tự NDAT)
* NMON: năm (tương tự NDAT)
* NNUM: dạng số (12 => 12)
* NTEL, NDIG: tương tự về cách đọc, đọc lần lượt từng số trong token (loại bỏ các ngăn cách như là ‘.’)
* NPER: thay ‘%’ thành ‘phần trăm’ còn phần số thì đọc như NNUM
* NSCR: dạng chuỗi các số (tỉ số là 2-3 => tỉ số là hai ba)
* NRNG: dạng khoảng cách, tương tự như NSCR nhưng thay ‘-’ thành ‘đến’
* NFRC: dạng phân số (1/2 => một phần hai). Ngoài ra các dạng đặc biệt như 24/7 => hai mươi bốn trên bảy
* NADD: dạng địa chỉ (số 14/3/2 phố huế => số mười bốn ngách 3 ngõ 2 phố huế), tùy số lượng của ‘/’ để thay đổi cách đọc

*4.2. NSWs dạng LETTERS*

*­*- Với các NSWs dạng LETTERS hầu như đã được xây dựng trong từ điển nên nếu không có trong từ điển thì tự xây dựng cách đọc theo từ.

- Các NSWs thuộc LWRD nếu như không đọc được thì tách từ theo các nguyên âm, phụ âm để đọc từng từ trong dạng này

- Cuối cùng nếu không đọc được thì đọc từng chữ cái trong từ đó =(((

*4.3. NSWs dạng OTHERS*

*­*- Diễn giải:

* PUNC: diễn giải theo từ điển (‘/’ => xẹt)
* DURA: không xử lý => hiển thị NSW để xử lý ở level sau
* URLE: [duc@gmail.com](mailto:duc@gmail.com) => dê u xê a còng gờ meo chấm com
* MONY: đọc phần số sau đó đọc phần tiền tệ
* CSEQ: đọc từng ký tự trong từ (15GM => một năm gờ mờ)
* NONE: loại bỏ NSW

***5. Bổ sung***

*5.1. Xây dựng từ tiếng anh sang văn bản tiếng việt (old)*

Với các từ tiếng Anh trong văn bản tiếng Việt, cụ thể là trong các bài báo, cần chuyển chúng về dạng đọc được của tiếng Việt. Các từ tiếng Anh sẽ được phân vào nhóm LWRD (đọc như một từ).

Phương pháp đề xuất:

* Xây dựng từ điển tương ứng giữa từ tiếng Anh và cách đọc tiếng Việt (1) cho một số từ thông dụng , kết hợp sử dụng từ điển đọc gần giống (2) để xử lý những từ ít phổ biến hơn.
* Với từ điển 1 sẽ có cấu trúc bao gồm nhiều dòng, mỗi dòng là một cặp từ tiếng Anh: cách đọc tiếng Việt.

Ví dụ: NATO : na tô

Paris : pa ri

London: Luân đôn

Napoleon : Na pô lê ông

* Tuy nhiên từ điển 1 sẽ không phủ được toàn bộ các từ tiếng Anh với cách đọc thường dùng. Vì vậy chúng ta cần sử dụng từ điển 2 với luật sinh cách đọc.

Chúng ta sử dụng từ điển CMU ([CMU pronouncing dictionary](http://www.speech.cs.cmu.edu/cgi-bin/cmudict)) để đưa ra được phiên âm tiếng Anh, sau đó chuyển về cách đọc tiếng Việt gần giống nhất. (có thể sử dụng thư viện pronouncing của python).

Ví dụ với thư viện CMU:

cheese => CH IY Z

read => R IY D

seizure => S IY ZH ER

Sau đó từ phiên âm tiếng Anh ta đưa về cách đọc tiếng Việt gần giống bằng cách dùng 1 số luật:

Ví dụ: IY chuyển thành i

ER chuyển thành ơ

S chuyển thành x,

SH chuyển thành s

ZH chuyển thành d

....

=> Cần xây dụng luật hoàn chỉnh, chú ý tới các ending sound, nếu không có trong tiếng Việt thì có thể chuyển đổi hoặc bỏ.

Ví dụ: cheese => CH IY Z có ending sound là Z thì ta bỏ

read => R IY D có ending sound là D thì ta chuyển thành t

Sau khi sử dụng luật để chuyển đổi, ta xây dựng được từ điển 2:

read => rit

seizure => xi dơ

*5.2. Xây dựng từ tiếng anh sang văn bản tiếng việt (new)*

Xây dựng một từ tiếng anh tương tự như cách cũ, tuy nhiên thay vì đưa về cách đọc tiếng việt ngay sau khi có phiên âm tiếng anh thì ta tách từ đã có phiên âm tiếng anh ra trước sau đó ta sẽ đưa về cách đọc tiếng việt để đọc từng từ chính xác hơn

Ví dụ: S IY ZH ER => S IY | ZH ER

Xử lý cách tách âm với một số dạng âm tiết đặc biệt như R

Ví dụ: Y UW0 K R EY1 N => Y UW0 | K | R EY1 N

Với từ điển để đưa về cách đọc sẽ được phân loại rõ ràng hơn với các âm được đánh trọng âm (Ví dụ AA có 3 cách đánh trong âm là AA0, AA1, AA2) và cách đọc âm theo từng vị trí ở đầu từ hay cuối từ.

Ví dụ: AE1 T => ât

HH AE1 T => hat

Tuy nhiên, trong quá trình test, cách đọc này không tốt hơn so với cách cũ nên vẫn đang sử dụng cách cũ. Vấn đề ở cách này là cách xây dựng thư viện cmu\_phones sao cho chính xác … \dicts\ cmu\_phones.txt