**Chuẩn hóa văn bản tiếng Việt**

**I. 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

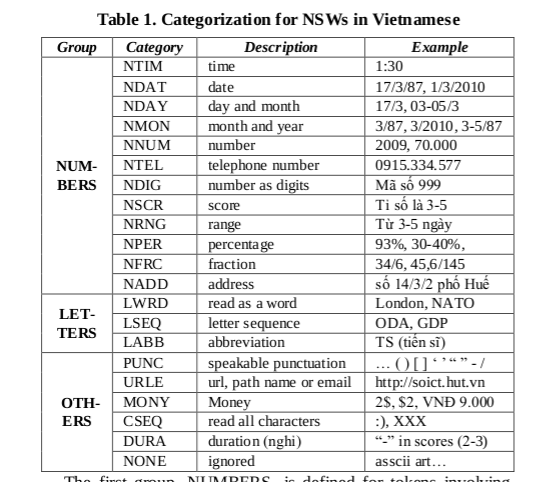
Các NSWs này không thể tìm ra cách phát âm theo luật “letter to sound”.

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

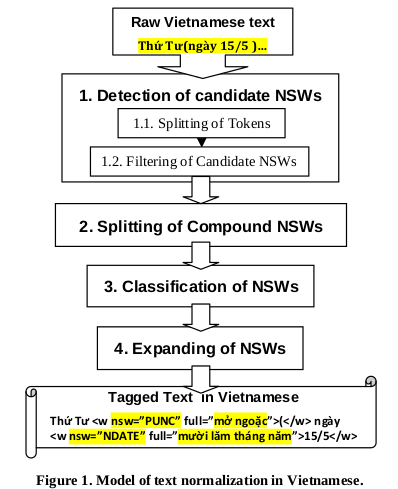
Ví dụ: “ngày 29/5” => “ngày hai mươi chín tháng năm”

PHUƠNG PHÁP: phân loại và chuẩn hóa từng loại

**II. Các loại NSWs**



**II. Mô hình chuẩn hóa**



**1. Detection of NSWs**

***1.1. Splitting of Tokens***

* Chia ra các từ tố ngăn cách bởi khoảng trắng.
* Tuy nhiên, cần nhóm các trường hợp đặc biệt:

Ví dụ: “09 15 33 45 77” => 09.15.33.45.77”

“150 000” => “150.000”

“$ 2000” => “$2000”

* Sửa các lỗi về khoảng trắng:

Ví dụ: “1. 000.000” => “1.000.000”

“2 / 3” => “2/3”

* Sửa các lỗi về dấu ngắt câu:

Ví dụ: “...đã xong.Viện CNTT ...”

”.. HN(Số 36) ..

* Chú ý đến Link URL, khi phân tách punctuation
* Các từ viết tắt dơn vị tiền tệ có thể ở sau hoặc trứớc số lượng:

Ví dụ:“S$2000” , “2000S$” , “2000SGD”

=> “hai nghìn đô la sing”

***1.2. Filtering of candidate NSWs***

Nếu token không khớp với các âm tiết trong từ điển thì sẽ được xem xét như 1 NSW, và được gán thẻ ‘w’. Các token còn lại gọi là token “thường” , được giữ lại để bước cuối cùng ghép nối lại với các token NSW.

Cần xem xét đến các dấu câu. Chúng được chia ra 2 loại:

+ loại 1: là loại không cần phát âm, bao gồm .,;:?!

=> giữ lại cho pha tiếp theo trong tổng hợp tiếng nói ở mức high-level

+ loại 2: các dấu còn lại, được xem là NSW trong nhóm PUNC.

=> gắn nhãn ‘w’

Ví dụ:

*“ Thứ, Tư, <w>(</w>, ngày, <w>15/5</w>,<w>)</w> vào*

*<w>8h30</w>, Trường, <w>ĐHBK</w>, <w>HN</w>, đã, đưa,*

*các, thông, tin, về, kỳ, tuyển, sinh, <w>ĐH</w>, năm, <w>2010</w>,*

*trên and <w>http://hut.edu.vn</w> ”*.

**2. Splitting of compound NSWs**

(Phân chia NSWs phức tạp)

* NSWs phức tạp là NSWs không thể phân loại vào các nhóm ở bảng 1, có thể phân chia ra các NSWs con. 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.

=> dùng biểu thức chính quy

* 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: “17/3/2010”, “100.000”, “0914.392.492”, “30-40%”...

* Các NSWs phức tạp được đặt trong cặp thẻ <split></split>

Ví dụ:

*“ Thứ, Tư, <w>(</w>, ngày, <w>15/5</w>, <w>)</w>, vào,*

*<split><w>8</w><w>h</w><w>30</w></split>, Trường,*

*<w>ĐHBK</w>, <w>HN</w>*

**3. Classification of NSWs**

Phân loại NSWs

Có nhiều phương pháp như cây quyết định, Bayes, mạng neural, SVM, ...

Các token ban đầu sẽ được chia ra thành 3 nhóm chính 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 thuộc tính nsw được cập nhật thành loại vừa được phân.

Ví dụ kết quả:

*“ Thứ, Tư, <w nsw=PUNC>(</w>, ngày, <w nsw=“NDATE”>15/5</w>,*

*<w nsw=PUNC>)</w>, vào, <split><w nsw=“NNUM”>8</w>*

*<w nsw=“LABB”>h</w> </split>, <w nsw=“NNUM”>30</w>,*

*Trường, <w nsw=“LABB”> ĐHBK</w>, <w nsw=“LABB”>HN</w>,*

*đã, đưa, các, thông, tin, về, kỳ, tuyển, sinh, <w nsw=“LABB”>ĐH</w>,*

*năm, <w nsw=“NNUM”> 2010</w>, trên, and <w nsw=”URLE”>*

*http://hut.edu.vn</w> ”*

***3.1. Phân loại nhóm NUMBERS***

Trong thực tế 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.

=> có thể sử dụng cây quyết định phân loại dựa vào ngữ cảnh và thuộc tính

định dạng. Ngữ cảnh bao gồm 2 token liền kề trước và 2 token liền kề sau.

Thuộc tính định dạng là số kí tự, số chữ số trong token.

***3.2. Phân loại nhóm LETTERS***

* Có 3 nhóm là LWRD, LSEQ, LABB. Phụ thuộc vào ngữ cảnh.

Ví dụ “NS” có thể thuộc từ viết tắt LABB cho “năm sinh”, cũng có thể thuộc nhóm chuỗi chữ cái LSEQ cho “nờ ét”

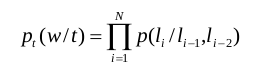
* Có thể dùng phương pháp thống kê:

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.

***3.3. Phân loại nhóm OTHERS***

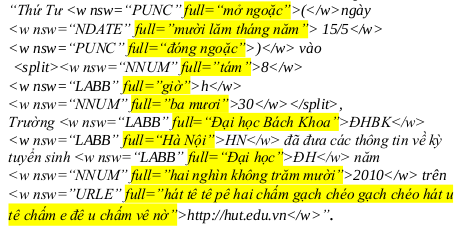
Mỗi loại lại có định dạng riêng biệt. Loại PUNC có thể đọc được bảo gồm dấu 3 chấm, các dấu mở đóng nháy đơn, nháy kép, ngoặc trong, ngoặc vuông, dấu nối, dấu xược (/). Loại URLE chứa @, <http://>, www, .com, .net,... Loại MONY chứa các kí hiệu tiền tệ $, VNĐ, S$,... trước và sau số lượng tiền. Loại CSEQ được gán là các kí tự khác, có thể đọc được, còn loại NONE là loại đặc biệt, không thể đọc được như biểu tượng ascii,...

**4. Expanding of NSWs**

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.

Ví dụ kết quả:



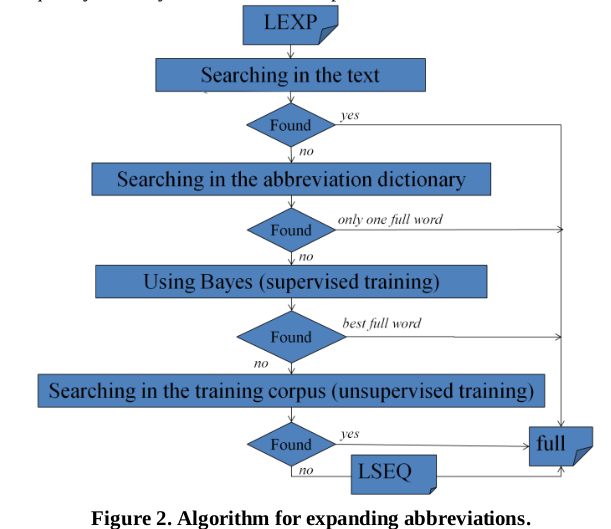
***4.1. Expanding of NSWs***

* NTIM trở thành số với “giờ” và “phút”, “giây” (1:03 một giờ ba phút)
* NDAT trở thành số với “tháng”, năm”. (02/03/98 hai tháng ba năm chín mươi tám)
* NDAY trở thành số với “tháng” (12/03 mười hai tháng ba)
* NMONT trở thành số với “năm” (12/1998 mười hai năm một nghìn chín trăm chín mươi tám)
* NNUM là số (12 mười hai)
* NTEL chuỗi các chữ số với các phân cách không được đọc (0386.151.514 không ba tám sáu một năm một năm một bốn)
* NDIG thành chuỗi các chữ số (3925 ba chín hai năm)
* NSCORE (tỉ số là 2-3 => tỉ số là hai ba)
* NRNG ”-” trở thành “đến” (từ 2-3 => từ hai đến ba)
* NPER “%” là phần trăm (30-40% => ba mươi đến bốn mươi phần trăm)
* NFRC phân số (3/4 ba phần tư)
* LWRD sử dụng từ (London => Luân Đôn)
* LSEQ đọc từng chữ cái (WTO vê kép tê ô)
* LABB sử dụng phương pháp mở rộng từ viết tắt
* PUNC đưa ra nghĩa đầy đủ ( ”...” => “ba chấm”)
* URLE [( van@gmail.com](mailto:(van@gmail.com) => vê a nờ a còng gờ meo chấm com)
* MONY kết hợp số và đơn vị tiền tệ, có thể thêm “/” thành “năm” (1$/người => một đô la mỗi người)
* CESQ đọc từng kí tự “:)” => “hai chấm mở ngoặc)
* DURA và NONE không cần mở rộng

4.2. Expanding of Abbreviations

Mở rộng từ viết tắt

Sử dụng từ điển, thuật toán như sau:



* Ban đàu tìm kiếm xem văn bản có dạng ” full Expanding (abbreviation)” không, nếu có thì đưa ra kết quả full luôn.

Ví dụ ” đại học Bách Khoa (ĐHBK)”

* Tìm kiếm trong từ điển, nếu chỉ có 1 full word thì đưa ra kết quả full. Nếu không thì sử dụng cả học có giám sát và không giám sát để tìm.
* Nếu cả hai đều không tìm được thì dùng LSEQ là đọc từng chữ cái

Ví dụ: KCB => ca xê bê

**III. Thử nghiệm**

Cần chuẩn bị:

* Từ điển “letter to sound”
* Từ điển từ viết tắt
* Bộ văn bản để train
* API cho Letter Language Model để đọc dữ liệu, tạo mô hình, tính entropy
* API cây quyết định bao gồm đọc data, sinh cây, định giá,...
* Dữ liệu được lấy từ các bài báo của các trang phổ biến.

**Phần 2:**

**Bổ sung xử lý các từ tiếng Anh trong văn bản tiếng việt**

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