**Chuẩn hóa văn bản tiếng Việt cho TTS**

**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

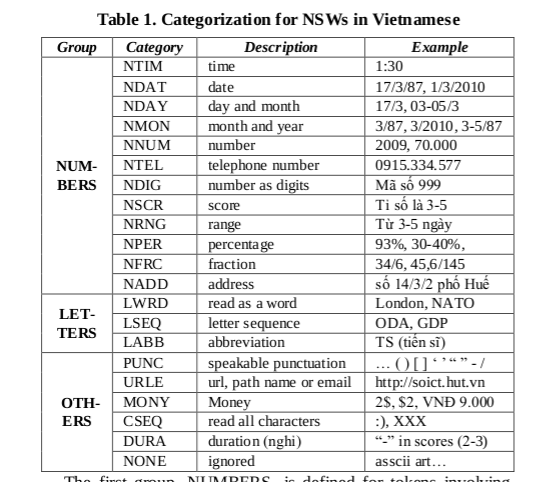
Các NSWs này không thể tìm ra cách phát âm theo luật “letter to sound”.

=> 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: (tạm thời) dùng regex

**II. Các loại NSWs**



**III. Mô hình chuẩn hóa**

| 1 | Tách token từ text |
| --- | --- |
| 2 | Tách các NSW phức tạp |
| 3 | Catch từng loại NSW qua regex và chuyển về dạng đã chuẩn hóa |

**1. Tách token từ text**

* Chuyển kí tự xuống dòng thành dấu chấm nếu cuối câu không có dấu chấm
* Tách các token ngăn cách bởi khoảng trắng
* Chỉnh sửa: tách từ và dấu câu dính nhau, tách các dấu câu liền nhau

Ví dụ: “đã xong.Viện CNTT” => “đã xong . Viện CNTT”

* Chỉnh sửa: Nhóm hoặc tách các trường hợp đặc biệt liên quan đến số

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

“150 000” => “150.000”

“$ 2000” => “2000$”

“2 , 63” => “2,63”

* Chuyển một số âm về cùng dạng, ví dụ “òa” => “oà” , “ụy” => “uỵ” ...
* Loại bỏ các dâu câu không cần đọc như (){}[] “” ‘’
* Trả về một mảng các tokens

**2. Tách các NSW phức tạp**

Các NSW phức tạp là các NSW không thể phân chia vào các nhóm ở bảng NSW. Thường thì các NSW phức tạp sẽ bao gồm cả số với chữ cái, hoặc với các dấu , ví dụ như 1m65, km/h,...

Cần tách chúng thành các thành phần trong bảng NSW, ví dụ 1m65 => 1 m 65

**3. Catch từng loại NSW qua regex và chuyển về dạng chuẩn hóa**

(kết hợp đếm số lần xuất hiện của từng loại NSW)

*Phương pháp tạm thời*: Thực hiện theo lần lượt, ưu tiên từ trên xuống dưới, chưa xét tới ngữ cảnh, nếu token nào khớp với regex thì thực hiện chuyển đổi sang fulltext. Đồng thời bộ đến sẽ tăng thêm cho loại NSW tương ứng.

*Thứ tự:*

* Chuyển URLE
* Chuyển ngày tháng năm (NDAT)
* Chuyển ngày tháng (NDAY)
* Chuyển tháng năm (NMON)
* Chuyển số kèm đơn vị (bao gồm NNUM và LABB), ví dụ 15 km / h
* Chuyển tiền tệ (MONY)
* Chuyển giờ (NTIM)
* Chuyển số điện thoại (NTEL)
* Chuyển phần trăm (NPER)
* Chuyển khoảng (NRNG) , ví dụ từ 3 - 4
* Chuyển phân số (NFRC)
* Hai từ nối nhau bởi dấu - thì bỏ dấu -, ví dụ Park Hang-seo
* Nếu token có trong từ điển tiếng Việt thì chuyển về chữ thường, nếu không thì xét bước tiếp theo
* Chuyển các trường hợp khác: (theo thứ tự trên xuống)
* Chuyển từ viết tắt nếu token có trong từ điển từ viết tắt LABB
* Nếu token in hoa thì chuyển theo LSEQ
* Nếu token có trong từ điển tiếng anh thì chuyển LWRD
* Nếu token là từ latin xử lý tên latin, vẫn thuộc lớp LWRD
* Chuyển số NNUM
* Chuyển 1 số dấu câu PUNC, ví dụ /
* Bỏ một số PUNC không đọc, ví dụ ()[}{
* Giữ lại không chuyển fulltext đối với các dấu ngắt nghỉ như .,;?! thuộc lớp DURA
* Loại bỏ các thành phần không đọc được như icon, các ngôn ngữ có kí tự không nằm trong charset tiếng Việt, tiếng Anh

=> *Đánh giá*: Do không sử dụng phân loại rõ ràng kết hợp với ngữ cảnh nên xảy ra tình trạng nhập nhằng.

*Các vấn đề*:

1. Ngày tháng, tháng năm được xét trước phân số. Do không xét đến ngữ cảnh nên một số trường hợp bị sai. Ví dụ “tỷ lệ là 15/11” => “tỷ lệ là mười lăm tháng mười một”
2. Xét đến khoảng NRNG trước nên tỷ số NSCR sẽ không được xét. Ví dụ: “tỷ số 2-3” => “tỷ số hai đến ba”
3. Với từ viết tắt, một từ viết tắt có thể có nhiều nghĩa, do không sử dụng ngữ cảnh nên chỉ sử dụng nghĩa đầu tiên có trong từ điển. Do đó một số trường hợp sẽ sai.
4. Trường hợp NDIG chưa được xét đến, do đưa về dạng NNUM. Ví dụ: “mã đăng nhập 635428” => “mã đăng nhập sáu trăm ba mươi lăm nghìn bốn trăm hai mươi tám”

**IV. XỬ LÝ CHO TỪNG LỚP**

1. NDIG

* Chuỗi các chữ số
* Tách và đọc từng chữ số

2. NNUM

* Đầu vào là số bao gồm cả . và , trong đó dấu phẩy để biểu diễn số thập phân, dấu chấm để phân tách hàng nghìn, hàng triệu, ...
* Xóa các dấu chấm phân tách
* Nếu có dấu phẩy (số thập phân ) thì phần nguyên đọc theo số thông thường, phần thập phân đọc từng số
* Phần nguyên sử dụng thư viện num2words để đọc

3. NTIM

* Là giờ:phút:giây hoặc giờ:phút
* Phân tách bởi dấu :
* Dùng NNUM đọc các số, kết hợp giờ, phút, giây

4. NDAT

* ngày/tháng/năm
* Phân tách bởi /
* Đọc ngày giống đọc số, chú ý thêm “mồng” trước ngày <10

5. NDAY

* ngày/tháng
* Tương tự NDAT

6. NMONT

* tháng/năm
* Tương tự NDAT

7. NTEL

* Là dãy số kết hợp dấu . thường bắt đầu bằng 0
* Bỏ dấu chấm
* Đọc từng chữ số

8. NSCR

* Là tỷ số, có dạng số-số
* Phân tách bởi dấu -
* đọc số thứ nhất rồi số thứ 2, không đọc dấu -
* Ví dụ: tỷ số 2-3 => tỷ số hai ba

9. NPER

* Là phần trăm hoặc một khoảng phần trăm
* Ví dụ 10% hoặc 10-15%
* Chuyển - thành “đến”, % thành “phần trăm”
* Đọc số dùng NNUM

10. NFRC

* Là phân số có dạng số/số
* Chuyển / thành “phần”
* Đọc số dùng NNUM

11. LWRD

* Các từ đọc thành tiếng, chủ yếu là từ tiếng Anh
* Xây dựng một từ điển popular\_english\_words gồm các từ tiếng anh phổ biến có dạng từ tiếng Anh|cách đọc tiếng Việt. Ví dụ: obama|ô ba ma
* Đối với các từ tiếng Anh không phổ biến, chưa có trong từ điển phía trên, ta xây dựng luật phiên âm tiếng Việt cho nó:
* Sử dụng từ điển CMU (có trong thư viện pronouncing) ta lấy được hàng trăm nghìn từ tiếng Anh và phiên âm CMU của nó
* Xây dựng từ điển âm tương ứng giữa tiếng Anh và tiếng Việt, ta được từ điển cmu\_phones có dạng âmCMU|âm tiếng Việt. Ví dụ ER|ơ, OW|âu
* Chuyển phiên âm tiếng Anh của 1 từ về chuỗi âm tiếng Việt. Ví dụ:

seizure => S IY ZH ER => xidơ

local => L OW C AH L => lâucơl

* Thực hiện tách các âm và ghép các âm với nhau thành các tiếng, tiến hành chỉnh sửa các âm về cách đọc tiếng Việt. Ví dụ:

ơl => ồ,

lâucơl=>lâu cồ,

taim => tham,...

* Thêm thanh điệu : at => át, ac => ác, ...
* Kết hợp từ điển các từ phổ biến phía trên và luật sinh phiên âm tiếngViệt cho từ điển CMU ta xây dựng được từn điển EN2VI\_DICT chứa các từ tiếng anh và phiên âm tiếng Việt của chúng

12. LSEQ

* Là chuỗi các kí tự,thường được viết hoa
* Ví dụ: ATM
* Xây dựng từ điển LSEQ\_DICT có dạng chữ cái|cách đọc, bao gồm các chữ cái tiếng Việt và tiếng anh. Ví dụ: A|a, T|tê, M|mờ
* Từ đó đưa ra cách đọc chuỗi bằng cách ghép cách đọc từng chữ

13. LABB

* Là các từ viết tắt của một cụm từ nào đó, thường được biểu diễn dưới dạng in hoa
* Ví dụ: NCKH=> nghiên cứu khoa học
* Xây dựng từ điển ABB\_DICT có dạng <từ viết tắt>|< các nghĩa được ngăn cách bởi dấu phẩy>
* Sau đó tra tra từ điển đưa ra từ đầy đủ, do chưa xét tới ngữ cảnh nên chỉ lấy nghĩa đầu tiên.

14. PUNC

* Là các dấu như /, ;...
* Xấy dựng từ điển dấu|cách đọc

15. URLE

* Là các link, email
* Chuyển các thành phần thường có sang dạng đọc được. Ví dụ: @=> a còng, gmail => gờ meo,...
* Chuyển số, dấu thành dạng đọc được
* Phân tách các thành phần ngăn cách bởi số, dấu câu
* Với mỗi thành phần phân tách thì: Xem nó có trong từ điển tiếng Việt không, nếu có thì chuyển về dạng chữ thường. Nếu không có thì xem nó có trogn từ điển tiếng Anh không, nếu có thì dùng từ điển chuyển sang cách đọc. Nếu không thì so khớp trong thành phần đó có từ nào có trong từ điển tiếng Việt không (ví dụ: vannguyen => thấy “van”, “nguy”, “en” có trong từ điển tiếng việt => đọc là “van nguy en”. Còn nếu còn thừa kí tự nào không so khớp thì đọc kí tự theo kiểu LSEQ.

16. MONY

* Là số gắn liền với đơn vị tiền tệ
* Đọc số, còn đơn vị thì xây dựng từ điển chuyển đổi cách đọc. Ví dụ: VNĐ|việt nam đồng,...

17. Tên latinh

* Là tên người, địa danh viết bằng chữ cái latinh a-z, tuy nhiên không có trong từ điển tiếng Anh (thuộc lớp LWRD)
* Thực hiện chuyển đổi một số âm. Ví dụ: al => an, j=>d, ic => ích
* Thực hiện so khớp từ xem có từ tiếng Việt nằm trong tên đó không, nếu có thì tách từ tiếng Việt ra, nếu không thì xét từng kí tự, nguyên âm thì đọc, phụ âm thì bỏ.
* Ví dụ: albeieqdadi =>anbâyeqđađi => an bây e đa đi.