A method for Vietnamese Text Normalization to improve the quality of speech synthesis

Thu-Trang Thi Nguyen
School of Information and
Communication Technology -
Hanoi University of Technology
1 Dai Co Viet, Hanoi, Vietnam
+84. (0)4 38.68.25.95
trangntt-fit@mail.hut.edu.vn

Thanh Thi Pham
School of Information and
Communication Technology -
Hanoi University of Technology
1 Dai Co Viet, Hanoi, Vietnam
+84. (0)4 38.68.25.95
Iambee2000@gmail.com

Do-Dat Tran
International Reseach Center MICA
CNRS UMI 2954 - Hanoi University of Technology
1 Dai Co Viet, Hanoi, Vietnam
+84 (0)4 38.68.30.87
Do-Dat.Tran@mica.edu.vn

ABSTRACT

Being necessary for a Text-To-Speech (TTS) system, text-normalization is general a challenging problem, especially for Vietnamese because of the local context. Recent researches in text-normalization in Vietnamese for TTS systems are still at the beginning with very simple sets of ad hoc rules for individual cases in spite of the ambiguity of real text. The purpose of this paper is to take some initial steps towards methodically normalizing input text in Vietnamese for a TTS system. This paper proposes a categorization and a normalization model for Vietnamese text based on related results for other languages. An experimental application is implemented to demonstrate the model, which uses several techniques including letter language model and decision trees for classifying NSWs and both supervised and unsupervised approaches for expanding abbreviations.

Categories and Subject Descriptors

Knowledge-based and information systems

General Terms

Standardization, Languages.

Keywords

Text Normalization, High-level Synthesis, Speech Synthesis, Pre-processing

1. INTRODUCTION

Developing systems that mimics human capabilities in generating speech for a range of human-machine communication has been a continuous expectation for many years. This field of study has known as speech synthesis, the "synthetic" (computer) generation of speech or Text-To-Speech (TTS), the process of converting written text into speech (1).

The input of real text for a TTS system is messy: numbers, dates, abbreviations, currency… are not standard words, called Non-Standard-Words – NSWs, in that one cannot find their pronunciation by applying "letter-to-sound" rules. Normalization of such words, called Text-Normalization, is the process of generating normalized orthography from text containing NSWs.

Text-Normalization is an essential requirement not only for TTS, but also for the preparation of training text corpora for acoustic-model and language-model construction (2). In this paper, we focus on Text-Normalization in Vietnamese which converses from the variety symbols, numbers, and other non-orthographic entities of text into a common orthographic transcription suitable for subsequent phonetic conversion.

Real text in Vietnamese often includes many non standard words (NSW) that one cannot find their pronunciation by an application of "letter-to-sound" in Vietnamese. Such NSWs include numbers; digit sequences (such as telephone numbers, date, time, codes…); abbreviations (such as ThS for "Thạc sĩ"); words, acronyms and letter sequences in all capitals (such as "GDP"); foreign proper names and place names (such as New York); roman numerals; URL's and email addresses…

There are already many researches on text normalization for different languages such as for English (3), for Hindi (4), for Japanese and Chinese (5), Greek (6), Bangla (7) or Indian (8). However, most of the proposed approaches are language specific as the problem depends on language properties.

Recently in Vietnam, TTS synthesis has received more serious attention. There are some researches/projects on this field such as Sao Mai Voice (9), "Hoa sung" (10), Voice of Southern Vietnam (11)... Unfortunately, text normalization in these projects is rarely considered to be interesting per se, let alone worthy of serious study. As a result, a result typical technology mostly involves sets of ad hoc rules tuned to handle on or two genres of text, with the expected result that the techniques, such as they are, do not generalize well to new domains. In addition to the ad hocity of most approaches, an additional problem with the lack of systematic work on the topic of NSWs, is that we do not have a clear idea of the range of types of NSWs that must be covered. Anyone who has worked on text normalization for a particular domain will be very familiar with the range of possible NSWs for that domain, and will have developed some ways of dealing with them, but there has been little or no work on developing a broader picture of the range of cases one could expect to find if one looks over a wider set of text types.

This paper proposes a method for text normalization in Vietnamese at addressing above problems. Based on results for other languages and Vietnamese properties, we have developed a categorization of NSWs. We then propose a model of text

SoICT'10, August 27–28, 2010, Hanoi, Vietnam.

normalization in Vietnamese. Finally, we have developed an experimental application follows the proposed NSW categorization and model of text normalization in Vietnamese. We have investigated some techniques to realize the application such as Letter Model Language (12) and Decision Tree (13) (14) for classifying NSWs and both supervised and unsupervised approaches for expanding NSWs.

This paper is organized as follows. Section 2 presents our categorization of NSWs in Vietnamese. In Section 3, we proposes a model for text normalization in Vietnamese with detail descriptions for activities in this model. Some preparations and results of the experiment is given in Section 4. We conclude and give some perspectives in Section 5.

2. NSW CATEGORIZATION

Since Vietnamese text is messy, it is necessary to have a thorough categorization for NSWs in Vietnamese. The different categories have been chosen to reflect anticipated differences in algorithms for transforming (or expanding) tokens to a sequence of words, where a "token" is a sequence of characters separated by white space (see on Section 3 for a more on defining tokens).

We have examined a variety of data in newspapers of "dantri.com.vn" because of their huge and diverse resource. Then we have developed a categorization for NSWs in Vietnamese, summarized in Table 1, to cover the different types of NSWs that we observed. There are three main groups including NUMBERS, LETTERS and OTHERS.

Table 1. Categorization for NSWs in Vietnamese

Group	***Category***	***Description***	***Example***
NUMBERS	NTIM	time	1:30
	NDAT	date	17/3/87, 1/3/2010
	NDAY	day and month	17/3, 03-05/3
	NMON	month and year	3/87, 3/2010, 3-5/87
	NNUM	number	2009, 70.000
	NTEL	telephone number	0915.334.577
	NDIG	number as digits	Mã số 999
	NSCR	score	Tỉ số là 3-5
	NRNG	range	Từ 3-5 ngày
	NPER	percentage	93%, 30-40%,
	NFRC	fraction	34/6, 45,6/145
	NADD	address	số 14/3/2 phố Huế
LETTERS	LWRD	read as a word	London, NATO
	LSEQ	letter sequence	ODA, GDP
	LABB	abbreviation	TS (tiến sĩ)
OTHERS	PUNC	speakable punctuation	… ( ) [ ] ' ' " " - /
	URLE	url, path name or email	http://soict.hut.vn
	MONY	Money	2$, $2, VNĐ 9.000
	CSEQ	read all characters	:), XXX
	DURA	duration (nghỉ)	"-" in scores (2-3)
	NONE	ignored	asscii art…

The first group, NUMBERS, is defined for tokens involving numbers. It includes the following categories:

– NTIM for time ("9:30" or "9.30")
– NDAT for date ("17/3/1987", "17/03/1987", "17/03/87")
– NDAY for day and month ("17/03" or "17/3")
– NMON for month and year ("03/87" or "3/87")
– NNUM for cardinal number ("200.000" or "200 000")
– NTEL for telephone number (*"38.68.39.39", "38 683 939"*)
– NDIG for sequence of digits (*"mã số 999"*).
– NSCR for score ("tỉ số 2-3")
– NRNG for a range of numbers ("từ 2-3 ngày" is normalized to "từ hai đến ba ngày").
– NPER for a percentage ("30-40%" is normalized to "ba mươi đến bốn mươi phần trăm").
– NFRC for fractions ("34/5")

Three categories are defined as the second group "LETTERS" for tokens that include only letters:

– LWRD for tokens that cannot applied the pronunciation rules such as foreign proper names or acronyms that are said as a word rather than a letter sequence ("NATO" is normalized to "na tô").
– LSEQ for tokens that are said as a letter sequence ("ODA" is normalized to "ô đê a").
– LABB for tokens that have to expanded using abbreviation expander ("PV" is expanded to "phóng viên").

The final group, "OTHERS", is defined for remain tokens:

– PUNC for speakable punctuations: ellipsises, quotation marks (' ', ""), parentheses (()) and square brackets ([]).
– URLE for URLs, path names or emails (http://hut.edu.vn)
– MONY for money: Varied not only in Vietnamese currency style ("2$", "5.000 VNĐ" or "5.000đ") but also in foreign currency styles ("$2", "VNĐ 5.000").
– CSEQ includes tokens, which we simply do not know to render at this point, should be spoken to all character sequence such as smiley face ":)" in email…
– DURA for a duration such as ".", "-" or " " in telephone number (0912.456.345) or "-" in scores (2-3).
– NONE for tokens that cannot be labeled in any appropriate category. It should be ignnore or be put in CSEQ label.

3. MODEL OF TEXT NORMALIZATION IN VIETNAMESE

NSW processing in Vietnamese is really complicated because of its variety and local properties. Many of sub-categories mentioned in Section 2 are pronounced according to principles that are quite different from the pronunciation of ordinary words or names such as "VTV" is spoken as "vê tê vê" while "NASA" is spoken as "na sa"; "Ronaldo" is spoken as "rô nan đô" while "pê-đan" is spoken as "pê (nghỉ) đan"...

Further more, there is a high degree of ambiguity in pronunciation (higher than for ordinary words) so that many items have more than one plausible pronunciation, and the correct one must be disambiguated from context. A very typical case is a number, which should be identified as a number or a string of digits such as "2010" could be "hai nghìn không trăm mười" as a number or a year/date or a number in an address, or "hai không một không" as a code or a telephone number. In addition, abbreviations also can be ambiguous such as "ĐT" can be expanded to "Điện thoại" as a telephone, or "Đội tuyển" (in "ĐT Việt Nam"...) as a selected team in a competition…

Because of the complication and ambiguity in Vietnamese, we cannot neither use only a simple set of ad hoc rules nor ignore the context of the input. In this section, we propose the model for text normalization with a method that can "study" based on the context in which NSWs appear.

Figure 1 presents the proposed model of text-normalization in Vietnamese based on related researches (3) (4) (7) for other languages as well as local context and typical properties of Vietnamese language.

In each step, the input text will be XML-based tagged respectively in order to provide classification information and pronounceable words (called full words) for NSWs without losing any information from the original text. We study and apply the most popular of tagging conventions (3) into our work. Each token identified as a NSW is given the tag *"w"* with an attribute *"nsw"* for category, an attribute *"full"* for the full word and a value as given by the labelers.

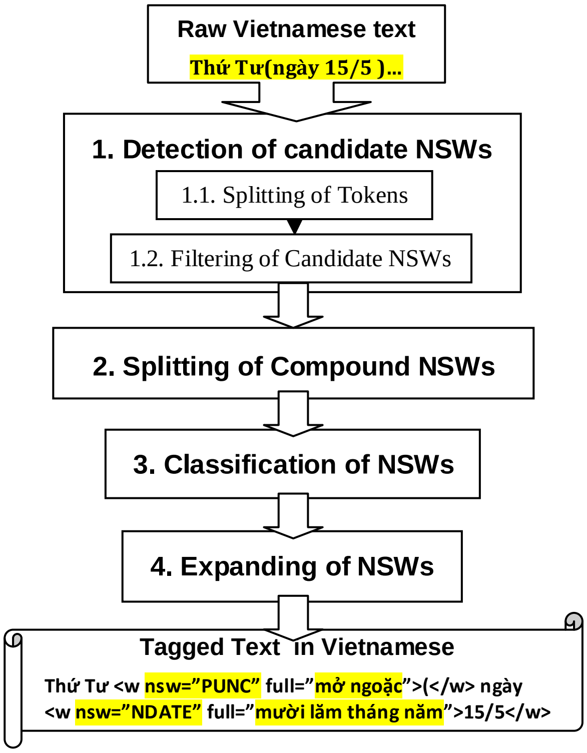
Raw Vietnamese text
Thứ Tư(ngày 15/5)...
1. Detection of candidate NSWs
1.1. Splitting of Tokens
1.2. Filtering of Candidate NSWs
2. Splitting of Compound NSWs
3. Classification of NSWs
4. Expanding of NSWs
Tagged Text in Vietnamese
Thứ Tư <w nsw="PUNC" full="mở ngoặc">(</w> ngày
<w nsw="NDATE" full="mười lăm tháng năm">15/5</w>

Figure 1. Model of text normalization in Vietnamese.

Firstly, in this model, input text in Vietnamese is passed through a process of detecting candidate NSWs by comparing whitespace-separated tokens with syllables in a dictionary including only pronounceable syllables in Vietnamese to get candidate NSWs, which do not appear in the dictionary. In the second steps, compound NSWs then are split into sub-tokens. Detected NSWs then are classified based on the proposal of NSW Categorization for Vietnamese in Section 2 using regex for simple cases; decision tree and letter language model for ambiguous cases. Finally, NSWs are expanded to the full format based on algorithms for each categories in each group. Language model is used in this step to disambiguate the output. In both NSWs Classification and NSWs Expanding steps, we use both unsupervised training and supervised training method.

To demonstrate the model, let us give a simple example with the input text "*Thứ Tư(ngày 15/5) vào 8h30,Trường ĐHBK HN đã đưa các thông tin về kỳ tuyển sinh ĐH năm 2010 trên http://hut.edu.vn*".

3.1 Step 1. Detection of NSWs

The purpose of this step is to identify candidate NSWs from the input text by using a dictionary that includes pronounceable syllables. Before filtering through the dictionary, the text has to be broken up into whitespace-separated tokens (syllables).

3.1.1 Step 1.1. Splitting of Tokens

The purpose of this step is to break down the input text into whitespace-separator tokens.

Before splitting the text, we have to group some special cases which should not be treated as several tokens but only one token. We have to replace white-spaces with "." between digits in telephone numbers such as *"09 15 33 45 77"* replaced to *"09.15.33.45.77"*, or in numbers such as *"150 000 000"* replaced to *"150.000.000"* or in money such as "*S$ 2000"* replaced to *"S$2000"*). Another special case is placing wrong white-space positions in text representation such as *"1. 000.000"* instead of *"1.000.000"*, *"3 /5/ 87"* instead of "*3/5/87"* or *"2 -3"* instead of *"2-3"*. In this case, we should remove extra white-spaces.

Moreover, in the real text, there are many faults in representing paragraphs, such as a punctuation such as full stop or comma is placed directly before the next word (E.g. *"Ngày 1/9/2009 tại phòng C1-222, đại diện tập đoàn IBM toàn cầu đã đến thăm Trường ĐHBK HN.Viện CNTT&TT đã ...)"*)...

These faults will cause some mistakes in some cases when splitting tokens by white spaces. For example in above cases of the wrong position of the comma or parenthes, after splitting tokens by white spaces, *"HN.Viện"* or *"HN(Số"* will be a token. Moreover, some punctuations such as ellipsises(...), quotation marks (‘’, “”), parentheses (()) and square brackets ([]) should be treated as a NSW because of their meaning in the text. So before splitting the input text, we propose to place a single white space directly preceding and following all punctuations.

However, some special cases including urls, emails, path names (URLE group), compound numbers and money will have side effects when applying these rules, such as "100.000" will become "100 . 000", which will be split into 3 tokens: "100", "." and "000" – hence spoken as "một trăm chấm không không không", the same situation with "trangntt-fit@mail.hut.edu.vn", "http://ngonngu.net", "30-40%" or "17/5/2010". Therefore, we should place a single white space directly preceding and following all punctuations, except these special cases.

Above special cases are easily-identifiable formats. The URLE group has a typical property including "@", "http://", "www" or ".com", ".net"…; compound numbers includes both numbers and punctuations while money is quite complicated. Currency symbols ($, VNĐ, S$, SGD…) could be appeared before or after the amount of money such as "S$2000", "2000S$" or "2000SGD" – all can be considered as "hai nghìn đô la sing".

Because of quite-complete rules, all activities in this step could be realized mainly by using regex. The result for the example is the following tokens: *Thứ, Tư, (, ngày, 15/5,), vào, 8h30, Trường, ĐHBK, HN, đã, đưa, các, thông, tin, về, kỳ, tuyển, sinh, ĐH, năm, 2010, trên* and *http://hut.edu.vn*.

3.1.2 Step 1.2. Filtering of candidate NSWs

The text then is broken up into whitespace-separated tokens. Each token except some special cases is then compared with Vietnamese pronounceable syllables in a dictionary, which is automatically built based on Vietnamese pronunciation rules (15) (16) (17). If the token is not matched with any syllable in the dictionary, it will be considered as a NSW.

All tokens considered as a NSW will be tagged with “w” indicating a candidate NSW. From now, we only have to work with tokens tagged with *“w”*, called *NSW tokens* (e.g. *“15/5”*); other tokens, called *“normal” tokens* (e.g. *“Thứ”*), should be preserved for the final step to concatenate to NSW tokens with their tags, called *NSW tags* (e.g. *“<w>15/5</w>”*).

Some special cases, which do not need comparing with syllables in the dictionary, are two types of punctuations. The first one is punctuations that do not need speaking out including stops (.), commas (,), semicolons (;), colons (:), question marks (?) and exclamation marks (!) while the second one should be considered as a NSW in PUNC category. The first type of punctuations should be preserved for next phases in high-level speech synthesis while the second one should be tagged with “w” indicating NSWs.

The result for the example is the following tokens:

“*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>*”.

3.2 Step 2. Splitting of Compound NSWs

Compound NSWs, which cannot be classified to any categories in Section 2, need to be split into more than one sub-NSWs. Compound NSWs usually includes both digits and letters (e.g. “I5108”, “1m65”) or both letters and hyphens (-) or slashes (/) (e.g. “kw/h”, “pê-đan”) or in both lower and upper case. These typical properties of compound NSWs can easily identified by regex.

Other NSWs must be in group such as numbers (telephone numbers, dates, ranges, scores…) since many of the same cues which indicate a likely split point (e.g. stop, hyphens, slashes) also serve as joining points in certain common entities (e.g. “17/3/2010”, “100.000”, “0914.392.492”, “30-40%”…). They should be treated as separate categories with their own expanding, so they do not need to be split.

After identifying compound NSWs, the second phase of splitting is relatively straightforward. The most productive split points are the following: at transitions from lower to upper cases (e.g. “WinXP”); at transitions from digits to letters (e.g. “1m70”, “19h50”, “E530”); and at punctuation (e.g. “34/2010/NĐ-CP”, “pê-đan”, “m/s”). These compound NSWs are tagged by <split>, and then broken down into sub-tokens applying the above rules. These sub-tokens also are filtered with the syllable dictionary and tagged in case of NSWs.

The result for the example is the following tokens:

“*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>, …*

3.3 Step 3. Classification of NSWs

Classifier is a traditional problem in data mining area. There are several kinds of algorithm for classifier such as decision tree, Bayes, Neural network, SVM… In the process of observing the real text as well as study several techniques, we find that with each category of NSWs, we should use suitable techniques that will be presented details in the following subsections.

NSW tokens first are classified in one of 3 groups in Section 2, which may be OTHERS, NUMBERS or LETTERS, by their identifiable formats, and added an attribute *nsw=”OTHERS”*, *nsw=”NUMBERS”* or *nsw=”LETTERS”* correlatively. They are then classified into one category of each group using corresponding algorithms/techniques.

Each detected token is added an attribute *nsw=“NDATE”* or *nsw=“MONY”*, so forth respectively, indicating the category of the NSW. The result of this step for the example is

“*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>*”.

3.3.1 Classification of categories in NUMBERS group

For the NUMBERS group, each category has a typical format; hence we can use the format to classify the NUMBERS group.

However, there are some ambiguous cases such as *“3-5”* can be treated as *“ba năm”* in *“tỉ số 3-5”* for NSCR, or *“ba đến năm”* in *“từ 3-5”* for NRNG, or *“ba tháng năm”* in *“ngày 3-5”* for NDAT; *“1/82”* can be treated as *“một phần tám hai”* in *“chiếm 1/82”* or *“tỉ lệ 1/82”* for NFRC, or *“một năm tám mươi hai”* in *“tháng 1/82”, or “một trên 82”* in *“số nhà 1/82”*… In such cases, in spite of the same format, the context is different so the way to read the NSW is also different. Therefore, the category of a NSW depends on its context.

From two points of view of the format and context of a NSW, we propose to build the decision tree with properties for the NSW including format properties and context properties. Format properties are numbers of characters and numbers of digits while context properties include 2 directly preceding tokens and 2 directly following tokens of the NSW.

3.3.2 Classification of categories in of Letter group

For the LETTERS group, which consists of strings of alphabetic characters, there is no specific format for each category and the context also do not play an important role for classification as LWRD, LSEQ or LEXP category. There is no clear distinguishing between these categories in LETTERS group: “*NS*” may be the abbreviation of “*năm sinh*” in “*NS 1987*” for LEXP category but may be “nờ ét” in “*NS1987*” for LSEQ category; “*UNESCO*” may be classified as LSEQ or LEXP, but in fact it is classified as LWRD (“iu nét xì cô”).

Hence, categories in LETTERS group then are classified by a traditional way, which estimate the probability of a NSW classified in each category. The NSW will be labeled in the category whose probability is maximum. This probability is calculated as the following statistical formulation: The probability of assigning tag ***t*** to observe NSW ***w*** can be estimated using the familiar Bayes approach as:

$$p(t/w) = \frac{p_t(w/t) * p(t)}{p(w)}$$

where – $t \in \{ASWD, SEQN, EXPN\}$

– The probability $p_t(w/t)$ can be discribed by a trigram Letter Language Model (LLM) for predicting observations of a particular t.

$$p_t(w/t) = \prod_{i=1}^{N} p(l_i / l_{i-1}, l_{i-2})$$

where $n = (l_1, l_2 \dots l_n)$ is the observed string made up of n characters.

Such LLMs have been used earlier for applications such as text compression and estimation of language entropy. The language model used in the most widely adopted n-gram (in our case trigram) formulation.

- The probability $p(t)$ is the prior probability of observing the NSW tag t in the text.
- The probability of the observed text or the normalization factor is given by:

$$p(w) = \sum_t p_t(w/t) * p(t)$$

This model assigns higher probability to shorter tokens in comparison to longer ones. However, the probabilities $p(t/w)$ which are compared always correspond to the same token, compensating for the length factor.

3.3.3 Classification of categories in OTHERS group

For OTHERS group, there is a particular format for each category. NSWs of PUNC category should be speakable punctuation including ellipsis's (...), quotation marks (‘’, “”), brackets (round () and square []), hyphens (-), slashes (/) while NSWs of URLE category should consist “@”, “http://”, “www” or “.com”, “.net”, “.vn”… MONY category should in the format: currency symbols ($, VNĐ, S$, SGD…) before or after the amount of money.

CSEQ category is labeled for other characters, which can be spoken. NONE category is labeled for special cases, which should not be read, such as asscii art, formating junk…

3.4 Step 4. Expanding of NSWs

The purpose of this step is to generate full words for NSWs, which is already classified in the previous step.

For most of tags the expansion to a word sequence is algorithmic. That is although there may be possibly be some choices in the pronunciation, the expansion algorithm is deterministic and not dependent of the domain or context.

The <*w*> tag of each NSW is then added an attribute *full=“mười lăm tháng năm”* or *full =“Hà Nội”*, so forth respectively, indicating the full format of the NSW. After expanding NSWs to their full format, which can be applied “letter-to-sound” rule, we concatenate “normal” tokens to tagged NSWs following the standard reorientation of text, which is summarized from (18).

The final result of text normalization for the example is

“Thứ Tư <w nsw=“PUNC” full=“mở ngoặc”>(</w>ngày
<w nsw=“NDATE” full=”mười lăm tháng năm”> 15/5</w>
<w nsw=“PUNC” full=“đóng ngoặc”>)</w> vào
<split><w nsw=“NNUM” full=”tám”>8</w>
<w nsw=“LABB” full=”giờ”>h</w>
<w nsw=“NNUM” full=”ba mươi”>30</w></split>,
Trường <w nsw=“LABB” full=“Đại học Bách Khoa”>ĐHBK</w>
<w nsw=“LABB” full=“Hà Nội”>HN</w> đã đưa các thông tin về kỳ tuyển sinh <w nsw=“LABB” full=“Đại học”>ĐH</w> năm
<w nsw=“NNUM” full=”hai nghìn không trăm mười”>2010</w> trên
<w nsw=”URLE” full=”hát tê tê pê hai chấm gạch chéo gạch chéo hát u tê chấm e đê u chấm vê nờ”>http://hut.edu.vn</w>”.

3.4.1 Expanding of NSWs

Even within these algorithmic expanders some are more trivial than others. The tagged tokens are treated as follows.

- NTIM expanded as a numbers with hours, minutes and seconds as appropriate, such as “1:30” is expanded to “một giờ ba mươi phút”.
- NDAT expanded as cardinal numbers between “tháng”, “năm” such as “17/03/87” is expanded to “mười bảy tháng ba năm tám mươi bảy”.
- NDAY expanded as cardinal numbers between “tháng” such as “17/3” is expanded to “mười bảy tháng ba”.
- NMONT expanded as cardinal numbers between “năm” such as 2/2010 is expanded to “hai năm hai nghìn không trăm mười”, “tháng 2-3/2010” is expanded to “tháng hai đến tháng ba năm hai nghìn không trăm mười”.
- NNUM expanded to string of words representing the cardinal number. This covers integers and float.
- NTEL expanded as string of digits with silence for punctuation, for “/” expanded to “hoặc” such as “0383.881283/85” is expanded to “không ba tám ba tám tám một hai tám ba hoặc tám lăm”.
- NDIG expanded to string of words one for each digit such as “Mã số 3922” is expanded to “Mã số ba chín hai hai”.
- NSCORE expanded as string of digits such as “tỉ số là 2-3” is expanded to “tỉ số là hai ba”.
- NRNG expanded to string of words representing the cardinal number for each end, “-“expanded to “đến” such as “từ 2-3” is expanded to “từ hai đến ba”.
- NPER expanded to string of words representing the cardinal number for the amount, “%” expanded to “phần trăm”, “-” expanded to “đến” such as “30-40%” is expanded to “ba mươi đến 40 phần trăm”.
- NFRC expanded to representing the cardinal number for each end, “/” expanded to “phần” such as “chiếm 1/3” expanded to “chiếm một phần ba”.
- LWRD expanded to a corresponding pronounceable word in a dictionary such as “London” expanded to “Luân đôn”.
- LSEQ expanded to a list of words one for each letter such as “WTO” expanded to “vê kép tê ô”.
- LABB for tokens that have to expand using abbreviation expander. The expander for LABB, because it is the only expander that contains interesting content is discussed in a next subsection.
- PUNC expanded to corresponding full meaning in Vietnamese such as “…” can be expanded to “ba chấm”.
- URLE expanded as a character sequence such as “soict2010@it-hut.edu.vn” expanded to “ét o i xê tê …”.
- MONY expanded to string of words to say the number; deals with various currencies, “/” expanded to “mỗi” such as “1$/người” expanded to “một đô la mỗi người”.
- CSEQ expanded to a list of words one for each character such as “:)” expanded to “hai chấm mở ngoặc”.
- DURA and NONE expanded to no words.

3.4.2 Expanding of Abbreviations

Expanding of Abbreviations is an interesting problem. After studying and observing special cases, we propose an algorithm for expanding abbreviations illustrated in Figure 2.

Firstly, an abbreviation in LEXP category is search in the input text to get the definition for the abbreviation if any. Since the habit of text reprentation is *“... full Expanding (abbreviation)”* such as *“... Trường Đại học Bách Khoa (ĐHBK) ...”*, we can map full Expanding for the abbreviation after checking their corresponding commonality such as "Đại học Bách Khoa" can be the Expanding for "ĐHBK". This first activity can be done quickly but only suitable for some special case.

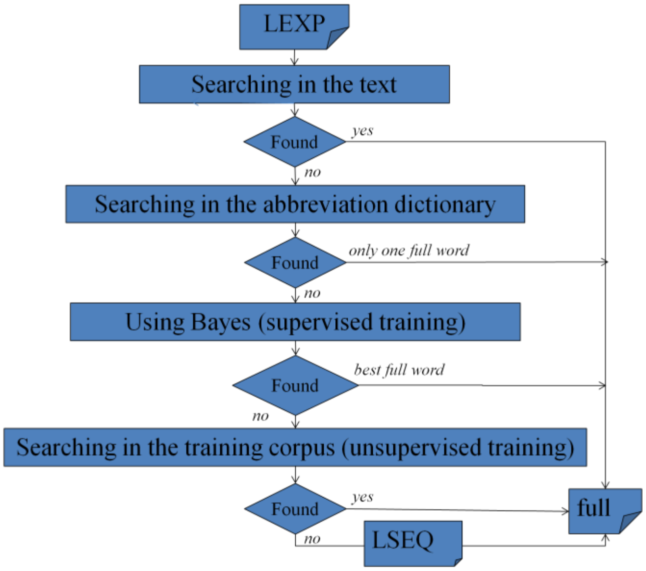
LEXP
Searching in the text
Found
yes
no
Searching in the abbreviation dictionary
Found
only one full word
no
Using Bayes (supervised training)
Found
best full word
no
Searching in the training corpus (unsupervised training)
Found
yes
full
no
LSEQ

Figure 2. Algorithm for expanding abbreviations.

If there is no Expanding for the abbreviation writen as the way of the first activity, the abbreviation is then looked up in the dictionary of abbreviations. This dictionary can be general or for a particular domain. The best case in this activity is that one can find only one full expanding for the abbreviations in a abbreviation dictionary. However, in fact, may be there is no expanding for the abbreviation or there are several ambiguous cases such as TS can be expanded to "Tiến sĩ" or "Thí sinh". Therefore, we propose to use both supervised (using Bayes) and unsupervised training method for ambiguous cases. In the worst case, we cannot give the answer for the expanding after all activities in the algorithm, we re-labeled the abbreviation as a LSEQ category of NSW, which is spoken as a sequence of letters constituting the abbreviation (such as if we cannot find the Expanding for "KCB" after all activities, we will re-label it to LSEQ and expand it to "ca xê bê").

In the supervised training method, we should have an appropriately tagged training corpus for a particular domain; one can compile a list of abbreviations from that corpus. The problem becomes:

"There are a set of abbreviations $e=\{e_1,e_2,...e_n\}$ and a set of full words $f=\{f_1,f_2,...f_n\}$. It is necessary to find f so that $p(f/e)$ is maximum. The probability is estimated based on Bayes:

$$p(f/e)=\frac{p(e/f)*p(f)}{p(e)}$$

where $p(f/e)$ is maximum when $p(e/f)*p(f)$ is maximum since $p(e)$ is a constant when f changes.

- $p(e/f)=\prod_{i=1}^{n} p(e_i/f_i)$
- $p(f)=\prod_{i=1}^{n} f_i/f_{i-1},f_{i-2}$

Here we use Language Model (LM) for predicting $p(f)$. In our particular implementation, we use trigram language models with modified Kneser-Ney backoff (19).

In the unsupervised method, we assume that the following three things: an automatically inferred list of "clean" words from the corpus; an automatically inferred list of potential abbreviations from the corpus; and a procedure for aligning clean words and n-grams of clean words in the corpus with their potential abbreviations.

The first two items can be inferred using the classification described in Section 3.4. Given the first two items, in order to produce the alighment required by the third one, we need an abbreviation model that for a given full word or phrase, can predict the set of possible abbreviations.

For example, suppose we have the abbreviation BV in our corpus as in the example: "Khám tại BV Bạch Mai" and that elsewhere in the corpus we find a example like the following: "Khám tại bệnh viện Bạch Mai"

We would like to guess that "BV" be an abbreviation of "bệnh viện". We would also like to assign some nominal probability estimate for $p(f/e)$ - the probability, given that one wants to abbreviate, e.g. "bệnh viện", that one would do it as "BV".

The advantage of the supervised method is faster than the unsupervised method but there should be a map from the abbreviation to the full word in the training corpus. The unsupervised method if more flexible and context-based but it requires more time to search in the training corpus and one may be cannot find the Expanding of the NSW.

4. EXPERIMENT

This section presents the process to prepare and do the experiment to demonstrate the proposed model. The following subsections summarize our activities and results for experiment.

4.1 Preparation for Experiment

To do the experiment for the model, we have to prepare the following items:

- A dictionary including pronounceable syllables, which can be applied "letter-to-sound" rule, has been built automatically based on the pronunciation system in Vietnamese (15) (16) (17) (18).
- A dictionary including popular abbreviations with their full words has been built and ordered automatically to get a list of abbreviations and has been manually given full words for abbreviations.
- A training corpus which will be discussed details in the next subsection.
- Developing an API for Letter Language Model including methods to read data, make the model, calculate entropy…
- Developing an API for Language Model including methods to read data, make the model, calculate entropy…
- Developing an API for Decision Tree following ID3 Algorithm including methods to read data from a text file (containing attribute values), make a tree, estimate…
- With the training corpus, build data for the decision tree, the language model and the letter language model.

4.2 Building the Training and Test Corpus

4.2.1 Building the Training Corpus

We have built the training corpus for both supervised training and unsupervised training method. The input for the training corpus includes 2000 articles from “dantri.com.vn” e-newspaper. There are various columns such as “Thể thao” (sport), “Giải trí” (Entertainment) … with different properties. Table 2 summarizes the distribution percentage of columns in “dantri.com.vn”.

Table 2. Article distribution by columns of the training corpus in the “dantri.com.vn” e-newspaper

Column	*Percentage*
Tin tức – sự kiện (News – Events)	20%
Thế giới (World)	15%
Thể thao (Sport)	15%
Giáo dục – Khuyến học (Education - Study Encouragement)	10%
Tình yêu - giới tính (Love – Gender)	5%
Ô tô – xe máy (Car – Automobile)	5%
Giải trí (Entertainment)	5%
Sức khỏe (Health)	5%
Sức mạnh số (Digital Power)	5%
Kinh doanh (Business)	5%
Nhịp sống trẻ (Young Life)	5%
Chuyện lạ (Odd Business)	5%

There are 3 main steps to build the training corpus:

- *Step 1: Preparation*. Articles are collected by a crawler from Cazoodle company. Each article is defined by a <record> tag with “id” attribute for ordinal number. There are 2 sub-tags of the <record> tag: <url> referring the URL of the article and <info> referring the content of the article.
- *Step 2: Automatically Tagging*. Automatically tagging for NSWs following the proposed model for the content of each article.
- *Step 3: Manually Tagging*. After automatically tagging, wrong cases should be re-tagged by hand.

In fact, Step 2 and Step 3 are executed repeatedly for each 200 articles. The first time there is no training corpus, but from the second time 200 articles, which are tagged exactly by hand, will be consider as the training corpus. The training corpus will be accumulated step by step and finally become a complete training corpus.

Table 3. Statistical results of the training corpus

Item	Amount
Articles	2000
Syllables	1.358.000
NSWs (~10.5%)	142.000

4.2.2 *Preparation for the Test Corpus*

We have prepared 2 test corpuses: one is included in the training corpus; another one is new, not included in the training corpus as the following:

- Test 1: Randomly extract 200 articles from the training corpus
- Test 2: 200 new articles from “news.socbay.com”.

Table 4 illustrates the distribution for Training Corpus and for Test Corpus 1 and Test Corpus 2 by 3 main groups: NUMBERS, LETTERS and OTHERS. Table 5 summarizes the distribution for Training Corpus and for Test Corpus 1 and Test Corpus 2 for categories in NUMBERS group while Table 6 presents the distribution for categories in LETTERS group.

Table 4. NSW distribution by 3 main groups

Group	*Distribution for Training Corpus*	*Distribution for Test 1*	*Distribution for Test 2*
NUMBERS	63%	61%	60.7%
LETTERS	32%	33%	33.3%
OTHERS	5%	6%	6%

Table 5. NSW Distribution by categories in NUMBERS group

Category	*Distribution for Training Corpus*	*Distribution for Test 1*	*Distribution for Test 2*
NTIM	5.10%	6.00%	5.00%
NDAT	6.70%	7.50%	7.00%
NDAY	12.20%	13.18%	11.20%
NMON	6.50%	7.09%	6.00%
NNUM	45.40%	43.10%	47.40%
NTEL	8.20%	8.20%	9.10%
NDIG	8.40%	8.50%	8.30%
NSCR	2.56%	2.00%	1.00%
NRNG	3.20%	3.00%	3.37%
NADD	0.04%	0.03%	0.03%
NPER	0.9%	0.7%	0.7%
NFRC	0.8%	0.7%	0.9%

Table 6. NSW Distribution by categories in LETTERS group

Category	*Distribution for Training Corpus*	*Distribution for Test 1*	*Distribution for Test 2*
ASWD	53.55%	50.12%	55.50%
EXPN	22.65%	24.82%	21.52%
SEQN	23.80%	25.00%	22.98%

4.3 Results of Experiment

After careful preparation for the experiment, we have built a tool integrated components/items in the Subsection 4.1, illustrating the proposed model in Section 3.

In the proposed model, the accuracy mainly depends on the third (Classification of NSWs) and the fourth step (NSWs Expanding because the first two step is mainly based on the format, not based on the context, hence a little ambiguous. Therefore, in Subsections 4.3.1 and 4.3.2, we summarize the results of the last two steps: Classification of NSWs and NSWs Expanding, which include ambiguous cases.

4.3.1 *Experiment Result in Classification of NSWs*

Table 7 presents the accuracy results for 2 test corpuses of categories in NUMBERS group. There are 3 categories which has wrong classification for NSWs: NDAY, NMON and NNUM. Since they have the same format, the classification has to be done based on the context. However, if the context is not clear, the prediction cannot operate properly.

The accuracy for Test 1 and Test 2 is not different much because the context of numbers is not various.

Table 7. Experimental Accuracy in NUMBERS group

Category	*Accuracy for Test 1*	*Accuracy for Test 2*
NTIM	100.00%	100.00%
NDAT	100.00%	100.00%
NDAY	92.25%	90.20%
NMON	95.12%	94.43%
NNUM	99.28%	98.75%
NTEL	100.00%	100.00%
NDIG	100.00%	100.00%
NSCR	100.00%	100.00%
NRNG	100.00%	100.00%
NADD	93.23%	90.12%
NPER	100.00%	100.00%
NFRC	96.28%	94.20%
Average	**98.29%**	**97.92%**

For LETTERS group, the accuracy for each category is quite good, which represented in Table 8. There are some ambiguous cases since a NSW can be LEXP in an instance ("NS" will be "năm sinh" in "NS 2001"), but LSEQ in another instance ("NS" will be "nờ ét" in "NS2001").

Table 8. Experimental Accuracy in LETTERS group

Category	*Accuracy for Test 1*	*Accuracy for Test 2*
LWRD	95.65%	93.21%
LEXP	98.53%	96.12%
LSEQ	98.17%	96.54%
Average	**96.94%**	**94.60%**

The accuracy for Test 2 is lower than Test 1 since there are some new NSWs or new context in Test 2.

Table 9. Accuracy of 3 groups in Classification of NSWs

Group	*Accuracy for Test 1*	*Accuracy for Test 2*
Number	98.29%	97.92%
Letter	96.94%	94.60%
Other	100.00%	100.00%
Average	**97.94%**	**96.94%**

The accuracy of 3 groups in Classification of NSWs is summarized in Table 9.

4.3.2 Experiment results for NSWs Expanding

For NSWs Expanding, we do the experiment in both cases: using Language Model and not using Language Model, illustrated in Table 10. The result of the experiment that does not use LM is much lower than the one use LM since LM includes the context when estimating the probability to expand for a NSW.

In this step, the accuracy of Test 1 is equal to the accuracy of Test 2 in both cases. This means Test 1 has no new abbreviation rather than Test 2.

Table 10. Experiment result in NSWs Expanding

Method	*Accuracy for Test 1*	*Accuracy for Test 2*
Without LM	87.72%	84.65%
With LM	94.50%	93.42%

5. CONCLUSION

Our work in this paper is, we believe, a significant advance in the state of the art in text normalization, especially in Vietnamese. We have proposed a model of text normalization in Vietnamese. The input, "raw" text including NSWs, first is filtered to find out NSWs, compound NSWs are split to smaller NSWs then all NSWs are classified in each category we suggested, finally they are expanded to the full words to get the output, "refined" text in which NSWs are tagged with classification information and pronounceable words for each NSWs. In the model, we propose to use different techniques to disambiguate for each step in the model: Decision Tree and Letter Language Model in Classification of NSWs and Language Model (including both supervised training and unsupervised training method) in NSWs Expanding.

We have implemented the experiment to demonstrate the model in a particular test and training corpus with quite-good results. This work will be the initiation for methodical and systematical research on text normalization in Vietnamese. We hope that there will be a new perspective and more deeply researches for text normalization in Vietnamese in near future.

We continue improving the model to enhance the performance and the accuracy, especially diversifying resources and types of resource for experiment.

6. REFERENCES

1. **Taylor Paul.** *Text-To-Speech Synthesis.* s.l. : Cambridge University Press, 2009.

2. **Xuedong Huang, Alex Acero and Hsiao-wuen Hon.** *Spoken Language Processing, A guide to Theory, Algorithm, and System Development.* s.l. : Prentice Hall, 2001.

3. **Richard Sproat, Alan Black, Stanley Chen, Shankar Kumar, Mari Osten-dorf, and Christopher Richards.** Normalization of Non-StandardWords. *Computer Speech and Language, Volume 15, Issue 3.* July 2001, pp. 287-333.

4. *Hindi Text Normalization.* **K. Panchapagesan, Partha Pratim Talukdar, N. Sridhar Krishna, Kalika Bali, and A.G. Ramakrishnan.** Hyderabad, India : s.n., 19-22 December 2004. Fifth International Conference on Knowledge Based Computer Systems (KBCS).

5. *Non-Standard Word and Homograph Resolution for Asian Language Text Analysis.* **Craig Olinsky, and Alan W Black.** Beijing, China : s.n., 2000. Sixth International Conference on Spoken Language Processing (ICSLP 2000).

6. *Text Normalization for the Pronunciation of Non-Standard Words in an Inflected Language.* **Gerasimos Xydas, Georgios Karberis, and Georgios Kouroupertroglou.** Samos, Greece : s.n., May 5-8, 2004. 3rd Hellenic Conference on Artificial Intelligence (SETN04).

7. *Text Normalization System for Bangla.* **Firoj Alam, S. M. Murtoza Habib, and Mumit Khan.** Lahore, Pakistan : s.n., January 22-24, 2009. Conference on Language and Technology 2009 (CLT09).

8. *Text Processing for Text to Speech Systems in Indian Languages.* **Anand Arokia Raj, Tanuja Sarkar, Satish Chandra Pammi, Santhosh Yuvaraj, Mohit Bansal, Kishore Prahallad, and Alan W Black.** Bonn, Germany : s.n., 2007. 6th ISCA Speech Synthesis Workshop SSW6.

9. **Sao Mai Computer Center for the Blind.** Sao Mai Voice. *Trung tâm Tin học Vì người mù Sao mai (Sao Mai Computer Center for the Blind - SMCC).* [Online] http://www.saomaicenter.org/category/general-category/bộ-đọc-sao-mai.

10. **International Research Center MICA.** Hoa Sung. [Online] http://www.mica.edu.vn/tts.

11. **Speech Language Processing Group, HCMUS.** Tiếng nói Phương Nam (Voice of Southern Vietnam - VOS). *AILAB Laboratory, University of Science Ho Chi Minh city, Vietnam.* [Online] http://www.ailab.hcmus.edu.vn/slp/vos/default.aspx.

12. *Statistical Methods for Speech Recognition.* s.l. : MIT Press, Cambridge, 1997.

13. *Induction of decision trees.* **J.Ross Quinlan.** 1986, Machine Learning 1, Vol. 1, pp. 81-106.

14. *Simplifying Decision Trees.* **J. Ross Quinlan.** 1987, International Journal of Man-Machine Studies 27, Vol. 3, pp. 221-234.

15. **Đoàn Thiện Thuật.** *Ngữ âm tiếng Việt.* s.l. : NXB Đại học Quốc gia Hà Nội (Vietname National University Publisher), 1997.

16. **Trần Ngọc Dụng.** Cẩm Nang Ngữ Pháp Tiếng Việt (Vietamese Grammar Handbook). [Online] 2009. tinhhoavietnam.net.

17. **Mai Ngọc Chừ, Vũ Đức Nghiệu, and Hoàng Trọng Phiến.** *Cơ sở ngôn ngữ học và tiếng Việt (Linguitics Foundation and Vietnamese).* s.l. : Nhà xuất bản Giáo dục (Education Publication), 2000.

18. *Ngôn ngữ học và tiếng Việt (Linguistics, Vietnamese and more...).* [Online] http://ngonngu.net.

19. *Improved backing-off for n-gram language modeling.* **Kneser, Reinhard and Hermann Ney.** 1995. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. 1, pp. 181-184.

20. **Bell, T.C., J.G. Cleary and I. Witten.** *Text Compression.* s.l. : Prentice Hall, Englewood Cliffs, 1990.