HMM-based Vietnamese Text-To-Speech : Prosodic Phrasing Modeling, Corpus Design System Design, and Evaluation

Thi Thu Trang Nguyen

UNIVERSITÉ PARIS SUD

Comprendre le monde, construire l'avenir®

Limsi

Université Paris-Sud

Ecole Doctorale 427: Informatique de Paris-Sud
Laboratoire d'Informatique
pour la Mécanique et les Sciences de l'Ingénieur

Specialty : Computer Science

Doctor of Science

Defense on Thursday, 24 September 2015

by

Thi Thu Trang NGUYEN

HMM-based Vietnamese Text-To-Speech: Prosodic Phrasing Modeling, Corpus Design System Design, and Evaluation

Committee:

Advisors:	Christophe D'ALESSANDRO	Directeur de recherche CNRS (LIMSI)
	Do Dat TRAN	Professeur (Institut Polytechnique de Hanoi, Vietnam)
Reviewers:	Philippe MARTIN	Professeur émérite (Université Paris-Diderot 7)
	Yannis STYLIANOU	Professeur (Université de Crète, Grèce)
Examiner:	Laurent BESACIER	Professeur (Université Joseph Fourier, Grenoble)
	Sophie ROSSET	Directeur de recherche CNRS (LIMSI)

Groupe Audio et Acoustique
LIMSI-CNRS
Rue John von Neumann - Campus Universitaire
d'Orsay - Bât 508
F-91405 Orsay Cedex, France

ED 427 - Université Paris-Sud
UFR Sciences Orsay
Batiment 650 rue Noetzlin
91405 Orsay Cedex, France

This dissertation is dedicated to: My son Teddy,
who was six months when I started,
My parents and my husband
for their love, endless support and encouragement.

Acknowledgements

Foremost, I would like to express my most sincere and deepest gratitude to my thesis advisors *M. Christophe d'ALESSANDRO* (Directeur de Recherche at LIMSI-CNRS, France), *Prof. TRẦN Đỗ Đạt* and *Prof. PHẠM Thị Ngọc Yến* (MICA-CNRS, Vietnam) for their continuous support and guidance during my PhD program, and for providing me with such a serious and inspiring research environment. I am really graceful to *Christophe* for his excellent mentorship, caring, patience, and immense knowledge on Text-To-Speech (TTS). His advise helped me in all the time of research and writing of this thesis. He has also helped me much in completing the joint program administration, applying for scholarship *Excellence Eiffel*, and funding for traveling or conference. I am very thankful to *Prof. Đạt*, *M. Eric CASTELLI* and *Prof. Yến* for shaping my thesis at the beginning, for their supports in applying for scholarship *Évariste Galois*, and for their enthusiasm and encouragement. *Prof. Đạt* has substantially facilitated my PhD research, especially at the time I was a freshman on speech processing and TTS, with his valuable comments on Vietnamese TTS.

I am fortunate to have the opportunity to work with *Albert RILLIARD* (LIMSI). He has brought me great joy and crucial encouragement during my PhD. He has taught me various essential knowledge, such as prosody, statistical analysis, and perceptual evaluation. That has had a great impact on steering this thesis, leading to considerable results for my work. I am very grateful to Albert for his caring and advice on research, writing and presentation.

It is my pleasure to thank my thesis reviewers: *Prof. Philippe MARTIN* (Université Paris-Diderot 7), and *Prof. Yannis STYLIANOU* (Toshiba's Cambridge Research Laboratory) for accepting and spending their time on reading and giving valuable feedback on my thesis. I would also like to thank *Mme. Sophie ROSSET* (LIMSI), and *Prof. Laurent BESACIER* (LIG) for their acceptance to be in my defense committee.

I would like to thank *Prof. Jacqueline VAISSIÈRE* (LPP) for her caring and support during my first three-month internship in France as well as my PhD. I highly appreciate the opportunity to know and work with *M. Alexis MICHAUD* (MICA). I am sincerely indebted to Alexis for his suggestions and valuable comments on linguistics and writing.

I take this opportunity to extend my heartfelt gratitude to my dear friends and colleagues ; especially *Marc* for his constructive discussions and co-operation ; *Areti, Hảo, Chi, Hải Anh, Thuỳ* for their encouragement and enthusiasm in revising English for my dissertation ; and together with *Olivier, David, Samuel*, *anh Cường, Khoa, Diệp, anh Sơn, Xuân* for their supports and comments for my PhD, and for many fun and a friendly working ambiance at LIMSI and MICA. I wish to give thanks to students: *Lan, Thắng, Tùng* and the subjects for their efforts in conducting/participating the perception tests at MICA; to my Vietnamese friends in Paris: *Khánh, Ngọc Anh, Bình* for their enthusiastic supports in recording sessions at LIMSI, and *anh Bắc, Hiếu* for their helpful suggestions.

The present research would not have been feasible without financial supports from the *French government* with the two scholarships: *Évariste Galois* and *Excellence Eiffel*. I would

also like to acknowledge the funding from the Région Ile-de-France through the *FUI ADN-TR project (2011-2014)*, Vietnamese NAFOSTED fund for participating conferences.

I also take this opportunity to express my gratefulness to *Prof. Nicole BIDOIT*, Director and to *Stéphanie DRUETTA*, Assistant of the *Ecole Doctorale d'Informatique de Paris-Sud* for their supports during my research.

Last but not the least, I would like to dedicate this moment to *my son Teddy* and *my husband Chí*, who have given me much courage to accomplish this thesis, to *my parents* for their endless love and support during all my PhD.

Contents

Notations and Abbreviations

Notation / Abbreviation	Expansion	Explanation
A	Adjective	
ADT	ADjuncT	
ANOVA	Analysis Of Variance	
AP	Adjective Phrase	
C	subordinate Conjunction	
CALM	Causal-Anticausal Linear filter Model	
CART	Classification And Regression Tree	
CC	Coordinate Conjunction	
CD-HMM	Continuous Distribution HMM	
CFG	Context-free Grammars	
DFKI		German Research Center for Artificial Intelligence
DRT	Diagnosis Rhyme Test	
DSP	Digital Signal Processing	
E	prEposition	
EHMM		A labeler included in the festvox project (http://festvox.org/)
EM	Expectation Maximization	
F0		Fundamental Frequency
G2P	Grapheme-To-Phoneme	
GUI	Graphical User Interface	
H	Head	Head element of a syntactic phrase
HMM	Hidden Markov Model	
HTK	Hidden markov model ToolKit	A portable toolkit for building and manipulating hidden Markov models
HTS	HMM-based speech synthesis	
I	Interjection	
IoIT	Institute of Information Technology	
IPA	International Phonetic Alphabet	
J48		The Java implementation of the C4.5 algorithm
JAWS	Job Access With Speech	The world's most popular screen reader
L		Determiner

Notation / Abbreviation	Expansion	Explanation
LCA	Lowest Common Ancestor	
LMA	Log Magnitude Approximation	
LP	Linear Prediction	
LPCFG	Lexical Probabilistic Context-free Grammars	
LTAG	Lexicalized Tree-Adjoining Grammars	
M	nuMeral	
MARY (TTS)	Modular Architecture for Research on speech sYnthesis	
MEA-SYLDIC	MEAningful SYLlable DICtionary	A pronunciation dictionary including all meaningful Vietnamese syllables extracted from a huge raw text
MFCC	Mel Frequency Cepstral Coefficents	
ML	Maximum Likelihood	
MLSA	Mel Log Spectrum Approximation	
MOS	Mean Opinion Score	
MSD-HMM	Multi-Space probability Distribution HMM	
N	Noun	
NLP	Natural Language Processing	
NP	Noun Phrase	
Np	Pronoun	
NSW	Non-Standard Word	
Nu	Unit noun	
OBJ	OBJect	Primary object
OBJ2	OBJect 2	Secondary object
OBL	OBLique	
OCR	Optical Character Recognition	
PCFG	Probabilistic Context-free Grammars	
PDF	Probability Density Function	
POS	Part-Of-Speech	Word class or a lexical category
PP	Prepositional Phrase	
PRD	PReDicate	
PRO-SYLDIC	PROnounceable SYLlable DICtionary	A pronunciation dictionary including all pronounceable Vietnamese syllables
PSOLA	Pitch Synchronous OverLap and Add	
R	adveRb	
S		Main or independent clause
SAMPA	Speech Assessment Methods Phonetic Alphabet	
SBAR/SB		Subornidate or dependent clause
SPTK	Speech signal Processing ToolKit	
SSML	Speech Synthesis Markup Language	

Notation / Abbreviation	Expansion	Explanation
SUB	SUBject	
T		Auxiliary/modal words
TD-PSOLA	Time-Domain Pitch Synchronous OverLap and Add	
ToBI	Tones and Break Indices, a set of conventions for transcribing and annotating the prosody of speech	
TTS	Text-To-Speech	
UCP		A phrase including two or more head elements in different categories, connected by a coordinating conjunction
V	Verb	
VCL	Vietnam Lexicography Centre	
VDTO	Vietnamese Di-Tonophone and Others	The analysis corpus including VDTS and other recorded sentences
VDTS	Vietnamese Di-Tonophone Speech	The final training corpus designed for VTed
VEVA	VTed EVAluation tool	
VNSP	VNSpeechCorpus for synthesis	
VOS	Voice Of Southern vietnam	A Vietnamese TTS system, http://www.ailab.hcmus.edu.vn/
VP	Verb Phrase	
VSYL	Vietnamese SYLlable	A new designed corpus with 100% syllable coverage
VTed/VTED	Vietnamese TExt-to-speech Development system	
VTParser		An adopted Vietnamese parser using averaged perceptron and shift-reduce parsing algorithm
WEKA	Waikato Environment for Knowledge Analysis	A collection of machine learning algorithms for data mining tasks:
WinPitch		A Windows speech analysis program optimized for intonation research
X-SAMPA	Extended Speech Assessment Methods Phonetic Alphabet	
XML	eXtensible Markup Language	
XP		Unclassified phrases
Z		Bound morphemes

List of Tables

List of Figures

Lists of Media files

- The PhD student page, including thesis introduction, list of publications with soft-copies, demo voices, ect. is available at https://perso.limsi.fr/trangntt.
- The online demonstration of the VTED system is available at https://perso.limsi.fr/trangntt/online-demo.
- The demo voices are available at https://perso.limsi.fr/trangntt/demo-voices or at the "Demo voices" menu of the PhD student page. This webpage includes several samples for different perception tests:
  - MOS test
  - Intelligibility test
  - Tone intelligibility test
  - Pair-wise comparison test.

Chapter 1

Vietnamese Text-To-Speech: Current state and Issues

Contents

1.1 Introduction

Building systems that mimic human capabilities in understanding, generating or coding speech for a range of human-to-human and human-to-machine interactions has been increasingly expected for many recent years. One important task for obtaining such systems is to artificially produce the human speech. This field of study is known both as speech synthesis, i.e. the generation of synthetic speech, and Text-To-Speech (TTS), i.e. the conversion of written text to machine-generated speech. A TTS system is one that reads text out loud through the computer's sound card or other speech synthesis devices.

Vietnamese, the official language of Vietnam, is a tonal language, in which pitch is mostly used as a part of speech, changing the meaning of a word/syllable. Although Vietnamese TTS has been recently receiving a range of research on a number of synthesis techniques, there is a need for a complete and high-quality TTS system for this language with an appropriate corpus. The initial motivation of this work was to build a high-quality TTS system assisting Vietnamese blind people to access written text. The main objective of this research was then narrowed to build a high-quality TTS system with unlimited vocabulary. The Hidden Markov Model [1] (HMM-)based speech synthesis technique, a statistical parametric approach that will be discussed in this chapter, was chosen for developing a Vietnamese TTS system due to its predominance on general quality, footprint and robustness. The initial tasks to build a high-quality TTS system for Vietnamese were first outlined as the following:

- Studying the Vietnamese phonetics and phonology to discover the way to model the lexical tones in phonemes;
- Designing and recording a new corpus, which covers both phonemic and tonal contexts, for Vietnamese TTS systems;
- Proposing a novel prosodic model to improve the quality of a HMM-based TTS system for Vietnamese;
- Designing a complete architecture and a contextual feature set for, and building, an HMM-based Vietnamese TTS system;
- Designing, carrying out and analyzing various perceptual evaluations of synthetic voices with respect to the lexical tones.

This chapter presents the current state and issues in Vietnamese TTS, from which propositions of this research are given. Section 1.2 shows main applications and the basic architecture of TTS systems. This section also describes the two main current speech synthesis techniques: (i) Source/filter synthesizer, and (ii) Concatenative synthesizer. Statistical parametric and unit selection synthesis, the two prominent state-of-the-art speech synthesis techniques, are discussed in Section 1.3. Based on our initial motivation and their pros and cons, the HMM-based speech synthesis, one of the most well known technique in the statistical parametric approach, was chosen to develop VTED, a Vietnamese TTS system. In Section 1.4, some main characteristics of the Vietnamese language are introduced. Section 1.5 presents the current state of Vietnamese TTS, including existing TTS software applications in real-life as well as related research. Some discussions on main issues on Vietnamese TTS, which were considered as the final motivation of this work, are given in Section 1.6.

1. A statistical Markov model for representing probability distributions over sequences of observations. The system being modeled is assumed to be a Markov process with unobserved (hidden) states.

1.2 Text-To-Speech (TTS)

1.2.1 Applications of speech synthesis

Over the last few decades, speech synthesis or TTS has considerably drawn attention and resources from not only researchers but also the industry. This field of work has progressed remarkably in recent years, and it is no longer the case that state-of-the-art systems sound overtly mechanical and robotic. The concept of high quality TTS synthesis appeared in the mid eighties, as a result of important developments in speech synthesis and natural language processing techniques, mostly due to the emergence of new technologies. In recent years, the considerable advances in quality have made TTS systems more common in various domains and numerous applications.

It appears that the first real-life use of TTS systems was to support the blind to read text from a book and converting it into speech. Although the quality of these initial systems was very robotic, they were surprisingly adopted by blind people due to their availability compared to other options such as reading braille or having a real person do the reading (Taylor, 2009). Nowadays, there have been a number of TTS systems to help blind users to interact with computers. One of the most important and longest applications for people with visual impairment is a screen reader in which the TTS can help users navigate around an operating system. Blind people also widely have been benefiting from TTS systems in combination with a scanner and an Optical Character Recognition (OCR) software application that gives them access to written information (Dutoit and Stylianou, 2003). Recently, TTS systems are commonly used by people with reading disorder (i.e. dyslexia) and other reading difficulties as well as by preliterate children. These systems are also frequently employed to aid those with severe speech impairment usually through a voice output communication aid (Hawley et al., 2013). Handicapped people have been widely aided by TTS techniques in Mass Transit.

Nowadays, there exist a large number of talking books and toys that use speech synthesis technologies. High quality TTS synthesis can be coupled with "a computer aided learning system, and provide a helpful tool to learn a new language". Speech synthesis techniques are also used in entertainment productions such as games and animations, such as the announcement of NEC Biglobe [2] on a web service that allows users to create phrases from the voices of Code Geass [3] – a Japanese anime series. TTS systems are also essential for other research fields, such as providing laboratory tools for linguists, vocal monitoring, etc. Beyond this, TTS systems have been used for reading messages, electronic mails, news, stories, weather reports, travel directions and a wide variety of other applications.

One of the main applications of TTS today is in call-center automations where textual information can be accessed over the telephone. In such systems, a user pays an electricity bill or books some travel and conducts the entire transaction through an automatic dialogue system (Taylor, 2009)(Dutoit, 1997). Another important use of TTS is in speech-based question answering systems (e.g. Yahoo! 2009 [4]) or voice-search applications (e.g. Microsoft, 2009 [5], Google 2009 [6]), where speech recognition and retrieval system are tightly coupled. In such those systems, users can pose their information need in a natural input modality, i.e. spoken language and then receive a collection of answers that potentially address the information need directly. On smartphones, some typical and well-known voice interactive applications

2. http://www.biglobe.co.jp/en/
3. http://www.geass.jp/
4. http://answers.yahoo.com/
5. http://www.live.com
6. http://www.google.com/mobile

whose main component is multi-lingual TTS are Google Now, Apple Siri, AOL, Nuance Nina, Samsung S-Voice, etc. In these software applications, a virtual assistant allows users to perform a number of personalized, effortless command/services via a human-like conversational interface, such as authenticating, navigating menus and screens, querying information, or performing transactions.

1.2.2 Basic architecture of TTS

The basic architecture of a TTS system, illustrated in Figure 1.1, has two main parts (Dutoit and Stylianou, 2003) with four components (Huang et al., 2001). The first three – i.e. Text Processing, Grapheme-to-Phoneme (G2P) Conversion and Prosody Modeling – belong to the high-level speech synthesis, or the Natural Language Processing (NLP) part of a TTS system. The low-level speech synthesis or Digital Signal Processing (DSP) part – forth component – generates the synthetic speech using information from the high-level synthesis. The input of a TTS system can be either raw or tagged text. Tags can be used to assist text, phonetic, and prosodic analysis.

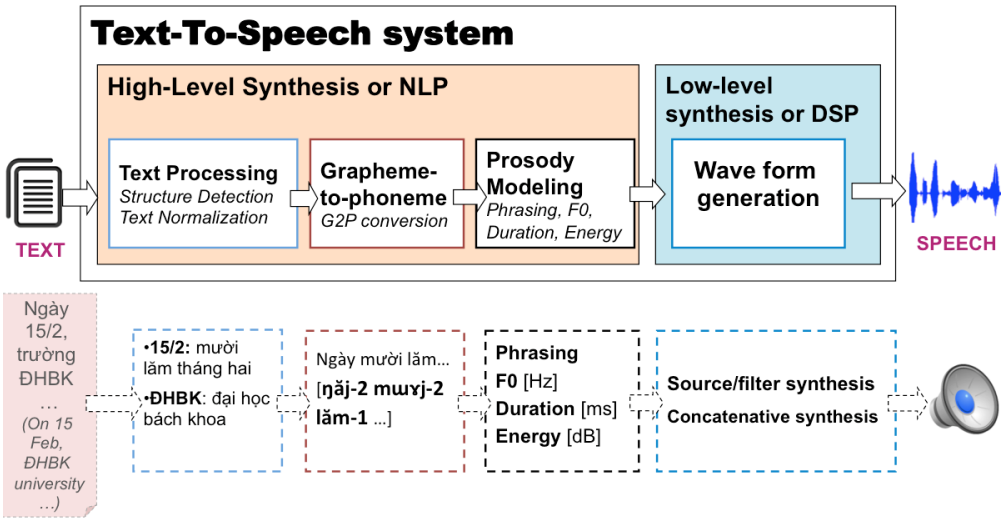


Figure 1.1 – Basic architecture of a TTS system (NLP: Natural Language Processing, DSP: Digital Signal Processing).

The Text Processing component handles the transformation of the input text to the appropriate form so that it becomes speakable. The G2P Conversion component converts orthographic lexical symbols (i.e. the output of the Text Processing component) into the corresponding phonetic sequence, i.e. phonemic representation with possible diacritical information (e.g. position of the accent). The Prosody Modeling attaches appropriate pitch, duration and other prosodic parameters to the phonetic sequence. Finally, the Speech Synthesis component takes the parameters from the fully tagged phonetic sequence to generate the corresponding speech waveform (Huang et al., 2001, p. 682). Due to different degrees of knowledge about the structure and content of the text that the applications wish to speak, some components can be skipped. For instance, some certain broad requirements such as rate and pitch can be indicated with simple command tags appropriately located in the text. An application that can extract much information about the structure and content of the text to be spoken considerably improve the quality of synthetic speech. If the input of the system contains the orthographic form, the G2P Conversion module can be absent. In some cases,

an application may have F0 contours pre-calculated by some other process, e.g. transplanted from a real speaker's utterance. The quantitative prosodic controls in these cases can be treated as "special tagged field and sent directly along with the phonetic stream to speech synthesis for voice rendition" (Huang et al., 2001, p. 6).

Text Processing. This component is responsible for "indicating all knowledge about the text or message that is not specifically phonetic or prosodic in nature". The basic function of this component is to convert non-orthographic items into speakable words. This called text normalization from a variety symbols, numbers, dates, abbreviations and other non-orthographic entities of text into a "common orthographic transcription" suitable for next phonetic conversion. It is also necessary to analyze white spaces, punctuations and other delimiters to determine document structure. This information provides context for all later processes. Moreover, some elements of document structure, e.g. sentence breaking and paragraph segmentation, may have direct implications for prosody. Sophisticated syntax and semantic analysis can be done, if necessary, for further processes, e.g. to gain syntactic constituency and semantic features of words, phrases, clauses, and sentences (Huang et al., 2001, p. 682).

G2P Conversion. The task of this component is to "convert lexical orthographic symbols to phonemic representation" (i.e. phonemes - basic units of sound) along with "possible diacritic information (e.g. stress placement)" or lexical tones in tonal languages. "Even though future TTS systems might be based on word sounding units with increasing storage technologies, homograph disambiguation and G2P conversion for new words (either true new words being invented over time or morphologically transformed words) are still necessary for systems to correctly utter every word. G2P conversion is trivial for languages where there is a simple relationship between orthography and phonology. Such a simple relationship can be well captured by a handful of rules. Languages such as Spanish and Finnish belong to this category and are referred to as phonetic languages. English, on the other hand, is remote from phonetic language because English words often have many distinct origins". Letter-to-sound conversion can then be done by general letter-to-sound rules (or modules) and a dictionary lookup to produce accurate pronunciations of any arbitrary word. (Huang et al., 2001, p. 683).

Prosody Modeling. This component provides prosodic information (i.e. "an acoustic representation of prosody") to parsed text and phone string from linguistic information. First, it it necessary to break a sentence into prosodic phrases, possibly separated by pauses, and to assign labels, such as emphasis, to different syllables or words within each prosodic phrase. The duration, measured in units of centi-seconds (cs) or milliseconds (ms), is then predicted using rule-based (e.g. Klatt) or machine-learning methods (e.g. CART). Pitch, a perceptual correlate of fundamental frequency (F0) in speech perception, expressed in Hz or fractional tones (semitones, quarter tones...), is generated. F0, responsible for the perception of melody, is probably the most characteristic of all the prosody dimensions; hence generation of pitch contours is an incredibly complicated language-dependent problem. Intensity, expressed in decibels (dB), can be also modeled. Besides, prosody depends not only on the linguistic content of a sentence, but also on speakers and their moods/emotions. Different speaking styles can be used for a prosody generation system, and different prosodic representations can then be obtained (Huang et al., 2001).

Speech Synthesis. This final component, a unique one in the low-level synthesis, takes predicted information from the fully tagged phonetic sequence to generate corresponding speech waveform. In general, there currently have been two basic approaches concerning speech synthesis techniques: (i) Source/filter synthesizers: Produce "completely synthetic" voices using a source/filter model from the parametric representation of speech, (ii) Concate-

native synthesizers: Concatenate pre-recorded human speech units in order to construct the utterance. The first approach has issues in generating speech parameters from the input text as well as generating good quality speech from the parametric representation. In the second approach, signal processing modification and several algorithms/strategies need to be employed to make the speech sound smooth and continuous, especially at join sections. Details on these approaches are presented in next subsections.

1.2.3 Source/filter synthesizer

The main idea of this type of synthesizer is to re-produce the speech from its' parametric representation using a source/filter model. This approach makes use of the classical acoustic theory of speech production model, based on vocal tract models. "An impulse train is used to generate voiced sounds and a noise source to generate obstruent sounds. These are then passed through the filters to produce speech" (Taylor, 2009, p. 410).

It turns out that formant synthesis and classical Linear Prediction (LP) are basic techniques in this approach. *Formant synthesis* uses individually "controllable formant filters which can be set to produce accurate estimations of the vocal tract transfer function". The parameters of the formant synthesizer are determined by a set of rules which examine the phone characteristics and phone context. It can be shown that very natural speech can be generated so long as the parameters are set very accurately. Unfortunately it is extremely hard to do this automatically. The inherent difficulty and complexity in designing formant rules by hand has led to this technique largely being abandoned for engineering purposes. In general, formant synthesis produces intelligible, often "clean" sounding, but far from natural. The reasons for this are: (i) the "too simplistic" source model, (ii) the "too simplistic" target and transition model, which misses many of the subtleties really involved in the dynamics of speech. While the shapes of the formant trajectories are measured from a spectrogram, the underlying process is one of motor control and muscle movement of the articulators (Taylor, 2009, p. 410). *Classical Linear Prediction* adopts the "all-pole vocal tract model", which is similar to formant synthesis with respect to the source and vowels in terms of production. It differs in that all sounds are generated by an all-pole filter, whereas parallel filters are common in formant synthesis. Its main strength is that the vocal tract parameters can be determined automatically from speech. Despite its ability to faithfully mimic the target and transition patterns of natural speech, standard LP synthesis has a significant unnatural quality to it, often impressionistically described as "buzzy" or "metallic" sounding. While the vocal tract model parameters can be measured directly from real speech, an explicit impulse/noise model can still be used for the source. The buzzy nature of the speech may be caused by an "overly simplistic" sound source (Taylor, 2009, p. 411).

The main limitations of those techniques concern "not so much the generation of speech from the parametric representation, but rather the generation of these parameters from the input specification which is created by the text analysis process. The mapping between the specification and the parameters is highly complex, and seems beyond what we can express in explicit human derived rules, no matter how "expert" the rule designer" (Taylor, 2009, p. 412). Furthermore, acquiring data is fundamentally difficult and improving naturalness often necessitates a considerable increase in the complexity of the synthesizer. The classical linear prediction technique can be considered as "a partial solution to the complexities of specification to parameter mapping", where the issue of generating of the vocal tract parameters explicitly is bypassed by data measurement. The source parameters however, are still "specified by an explicit model, which was identified as the main source of the unnaturalness" (Taylor, 2009, p. 412).

A new type of glottal flow model, namely a Causal-Anticausal Linear filter Model (CALM), was proposed in the work of Doval et al. (2003). The main idea was to establish a link between two approaches of voice source modeling, namely the spectral modeling approach and the time-domain modeling approach, that seemed incompatible. Both approaches could be envisaged in a unified framework, where time-domain models can be considered, or at least approximated by a mixed CALM. The "source/filter" model can be considered as an "excitation/filter" model. The non-linear part of the source model is associated to the excitation (i.e. quasi-periodic impulses), and the mixed causal-anticausal linear part of the model is associated to the filter component, without lack of rigor.

1.2.4 Concatenative synthesizer

This type of synthesizer is based on the idea of concatenating pieces of pre-recorded human speech in order to construct a utterance. This approach can be viewed as an extension of the classical LP technique, with a noticeable increase in quality, largely arising from the abandonment of the over simplistic impulse/noise source model. The difference of this idea from the classical linear prediction is that the source waveform is generated using templates/samples (i.e. instances of speech units). The input to the source however is "still controlled by an explicit model", e.g. "an explicit F0 generation model of the type that generates an F0 value every 10ms" (Taylor, 2009, p. 412).

During database creation, each recorded utterance is segmented into individual phones, di-phones, half-syllables, syllables, morphemes, words, phrases or sentences. Different speech units considerably affect the TTS systems: a system that stores phones or di-phones provides the largest output range, but may lack clarity. For specific (limited) domains, the storage of entire words, phrases or sentences allows for high-quality output. However, di-phones are the most popular type of speech units, a di-phone system is hence a typical concatenative synthesis system.

The synthesis specification is in the form of a list of items, each with a verbal specification, one or more pitch values, and a duration. The prosodic content is generated by explicit algorithms, while signal processing techniques are used to modify the pitch and timing of the di-phones to match that of the specification. *Pitch Synchronous OverLap and Add (PSOLA)*, a traditional method for synthesis, operates in the time domain. It separates the original speech into "frames pitch-synchronously" and performs modification by overlapping and adding these frames onto a new set of epochs, created to match the synthesis specification. Other techniques developed to modify the pitch and timing can be found in the work of Taylor (2009).

While this is successful to a certain extent, it is not a perfect solution. It can be said that we can "never collect enough data to cover all the effects we wish to synthesize, and often the coverage we have in the database is very uneven. Furthermore, the concatenative approach always limits us to recreating what we have recorded; in a sense all we are doing is reordering the original data" (Taylor, 2009, p. 435). One other obvious issue is how to successfully join sections of a waveform, such that the joins cannot be heard hence the final speech sounds smooth, continuous and not obviously concatenated. The quality of these techniques is considerably higher than classical, impulse excited linear prediction. All these have roughly similar quality, meaning that the choice of which technique to use is mostly made of other criteria, such as speed and storage.

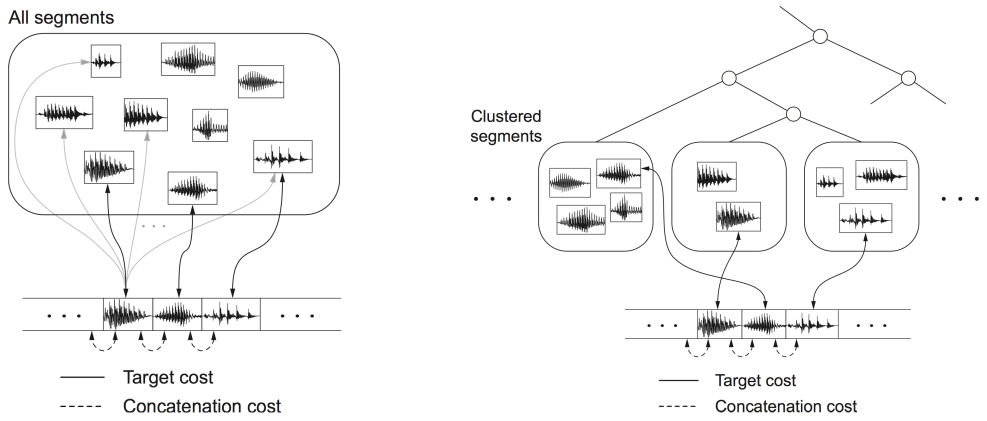
1.3 Unit selection and statistical parametric synthesis

Based on two basic approaches of speech synthesis, many improvements have been proposed for a high-quality TTS system. Statistical parameter speech synthesis along with the unit selection techniques are termed two prominent state-of-the-art techniques and hence widely discussed by a number of researchers with different judgments. This section describes and makes a comparison of those techniques.

1.3.1 From concatenation to unit-selection synthesis

In a concatenative TTS system, the pitch and timing of the original waveforms are modified by a signal processing technique to match the pitch and timing of the specification. Taylor (2009, p. 474) made two assumptions for a di-phone system: (i) "within one type of di-phone, all variations are accountable by pitch and timing differences" and (ii) "the signal processing algorithms are capable of performing all necessary pitch and timing modifications without incurring any unnaturalness". It appears that these assumptions are "overly strong, and are limiting factors on the quality of the synthesis. While work still continues on developing signal processing algorithms, even an algorithm which changed the pitch and timing perfectly would still not address the problems that arise from first assumption. The problem here is that it is simply not true that all the variation within a di-phone is accountable by pitch and timing differences".

The observations about the weakness of concatenative synthesis lead to the development of "a range of techniques collectively known as unit-selection. These use a richer variety of speech, with the aim of capturing more natural variation and relying less on signal processing". The idea is that for each basic linguistic type, there are a number of units, which "vary in terms of prosody and other characteristics" (Taylor, 2009, p. 475).



(a) General unit-selection scheme (b) Clustering-based unit-selection scheme

Figure 1.2 – General and clustering-based unit-selection scheme: Solid lines represent target costs and dashed lines represent concatenation costs (Zen et al., 2009).

In the unit-selection approach, new naturally sounding utterances can be synthesized by selecting appropriate sub-word units from a database of natural speech (Zen et al., 2009), according to how well a chosen unit matches a specification/a target unit (i.e. target cost) and how well two chosen units join together (i.e. concatenation cost). During synthesis, an algorithm selects one unit from the possible choices, in an attempt to find the best overall sequence of units that matches the specification (Taylor, 2009). The specification and the

units are entirely described by a feature set including both linguistic features and speech features. A Viterbi style search is performed to find the sequence of units with the lowest total cost, which is calculated from the feature set.

According to the review of Zen et al. (2009), there seem to be two basic techniques in unit-selection synthesis, even though they are theoretically not very different: (i) the selection model (Hunt and Black, 1996), illustrated in Figure 1.2a (ii) the clustering method that allows the target cost to effectively be pre-calculated (Donovan et al., 1998), illustrated in Figure 1.2b. The difference is that, in the second approach, units of the same type are clustered into a decision tree that asks questions about features available at the time of synthesis (e.g., phonetic and prosodic contexts).

1.3.2 From vocoding to statistical parametric synthesis

As mentioned earlier, the main limitation of source/filter synthesizers is generating speech parameters from the input specification that was created by text analysis. The mapping between the specification and the parameters is highly complex, and seems beyond what we can express in explicit human derived rules, no matter how "expert" the rule designer is (Taylor, 2009). It is hence necessary a "complex model", i.e. trainable rules from speech itself, for that purpose.

The solution can be found partly from the idea of vocoding, in which a speech signal is converted into a (usually more compact) representation so that it can be transmitted. In speech synthesis, the parameterized speech is stored instead of transmitted. Those speech parameters are then proceeded to generate the corresponding speech waveform. As a result, the statistical parametric synthesis is based on the idea of vocoding for extracting and generating speech parameters. But the most important is that it provides statistical, machine learning techniques to automatically train the specification-to-parameter mapping from data, thus bypassing the problems associated with hand-written rules. Extracted speech parameters are aligned together with contextual features/features to build "trained models".

In a typical statistical parametric speech synthesis system, parametric representations of speech including spectral and excitation parameters (i.e. vocoder parameters, which are used as inputs of the vocoder) are extracted from a speech database and then modeled by a set of generative models. The Maximum Likelihood (ML) criterion is usually used to estimate the model parameters. Speech parameters are then generated for a given word sequence to be synthesized from the set of estimated models to maximize their output probabilities. Finally, a speech waveform is reconstructed from the parametric representations of speech (Zen et al., 2009).

Although any generative model can be used, HMMs have been particularly well known. In HMM-based speech synthesis [7] (HTS) (Yoshimura et al., 1999), the speech parameters of a speech unit such as the spectrum and excitation parameters (e.g. fundamental frequency - F0) are statistically modeled and generated by context dependent HMMs. Training and synthesis are two main processes in the core architecture of a typical HMM-based speech synthesis system, as illustrated in Figure 1.3 (Yoshimura, 2002).

In the training process, the ML estimation is performed using the Expectation Maximization (EM) algorithm, which is very similar to that for speech recognition. The main difference is that both spectrum (e.g., mel-cepstral coefficients and their dynamic features) and excitation (e.g., log F0 and its dynamic features) parameters are extracted from a database of natural speech modeled by a set of multi-stream context-dependent HMMs. Another differ-

7. http://hts.sp.nitech.ac.jp/

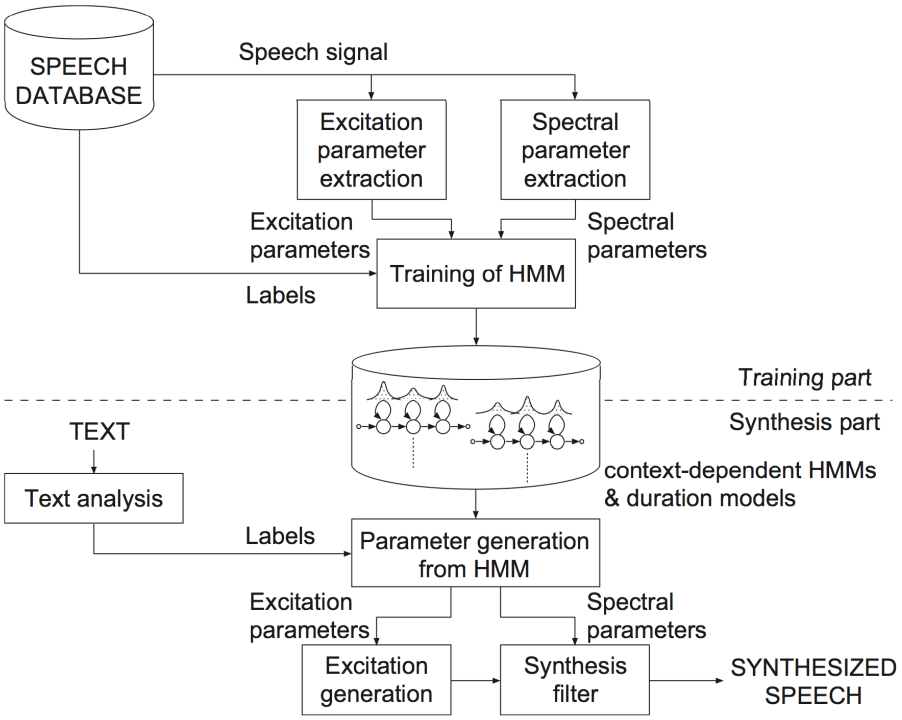


Figure 1.3 – Core architecture of HMM-based speech synthesis system (Yoshimura, 2002).

ence is that linguistic and prosodic contexts are taken into account in addition to phonetic ones (called contextual features). Each HMM also has its state-duration distribution to model the temporal structure of speech. Choices for state-duration distributions are the Gaussian distribution and the Gamma distribution. They are estimated from statistical variables obtained at the last iteration of the forward-backward algorithm.

In the synthesis process, an inverse operation of speech recognition is performed. First, a given word sequence is converted into a context-dependent label sequence, and then the utterance HMM is constructed by concatenating the context-dependent HMMs according to the label sequence. Second, the speech parameter generation algorithm generates the sequences of spectral and excitation parameters from the utterance HMM. Finally, a speech waveform is synthesized from the generated spectral and excitation parameters using excitation generation and a speech synthesis filter (Zen et al., 2009, p. 4), that is a vocoder with a source-excitation/filter model.

Figure 1.4 illustrates the general scheme of HMM-based synthesis (Zen et al., 2009, p. 5). In an HMM-based TTS system, a feature system is defined and a separate model is trained for each unique feature combination. Spectrum, excitation, and duration are modeled simultaneously in a unified framework of HMMs because they have their own context dependency. Their parameter distributions are clustered independently and contextually by using phonetic decision trees due to the combination explosion of contextual features. The speech parameter generation is actually the concatenation of the models corresponding to the full context label sequence, which itself has been predicted from text. Before generating parameters, a state sequence is chosen using the duration model. “This determines how many frames will be generated from each state in the model. This would clearly be a poor fit to real speech where the variations in speech parameters are much smoother”.

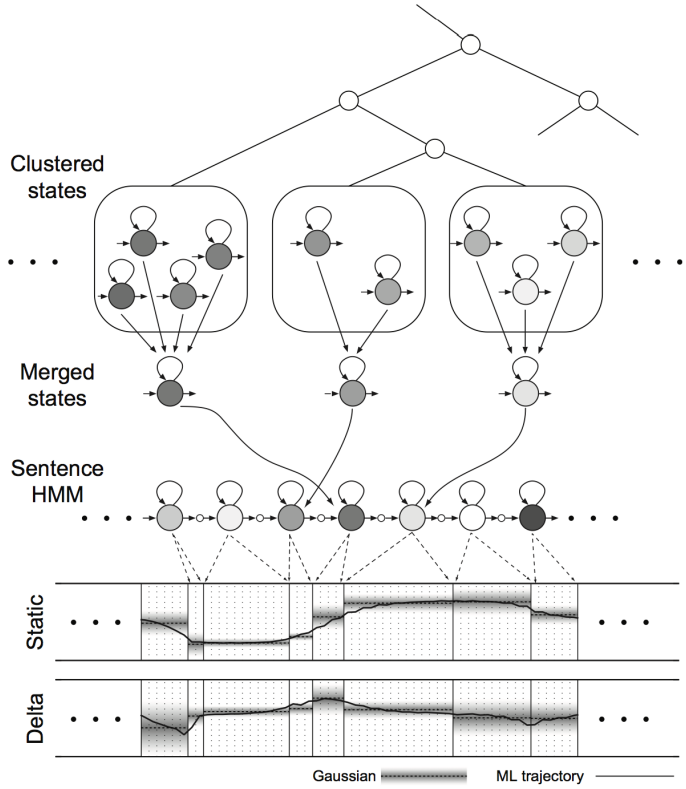


Figure 1.4 – General HMM-based synthesis scheme (Zen et al., 2009, p. 5).

1.3.3 Pros and cons

Statistical parameter speech synthesis offers an alternative to overcoming limitations of the parametric synthesis approach, which uses statistical machine learning techniques to infer the specification-to-parameter mapping from data. This technique can be simply described as "generating the average of some sets of similarly sounding speech segments". That directly contrasts with the purpose of unit-selection synthesis that "retains natural unmodified speech units, but using parametric models offers other benefits" (Zen et al., 2009). Unit selection speech synthesis is a sub-type and a natural extension of concatenative synthesis, and deals with the issues of "how to manage large numbers of units, how to extend prosody beyond just F0 and timing control, and how to alleviate the distortions caused by signal processing" (Taylor, 2009, p. 474). While both those techniques mainly depend on data, in the concatenative approach, the data is effectively memorized, whereas in the statistical approach, the general properties of the data are learned Taylor (2009, p. 447).

As mentioned above, while many possible approaches to statistical synthesis are possible, most work has focused on using hidden Markov models (HMMs). Main differences between unit-selection and HMM-based speech synthesis are summarized in Table 1.1.

Both approaches use features of speech units but in different ways. In the HMM-based speech synthesis, contextual features including phonetic, linguistic and prosodic features are used in both training and synthesis: (i) in training, contextual features are force-aligned with speech parameters to build context-dependent HMMs (ii) in synthesis, contextual features are used to build a context-dependent label sequence and according to that, an utterance HMM is constructed by concatenating the context-dependent HMMs. Whereas, the unit-selection

Table 1.1 – Unit-selection and HMM-based speech synthesis

Criteria	Unit-selection synthesis	HMM-based synthesis
Approach	Data-driven: memorize data (natural speech)	Parameter-driven: learn properties of data
	Multi-template	Statistics
Idea	Retain natural unmodified units by selecting appropriate sub-words	Generate the average of some sets of similarly sounding segments
Preferred applications	Limited domain	Open domain
Techniques	Target cost, concatenation cost	Machine learning
	Single tree	Multiple trees (spectral, F0, duration)
Quality	Discontinuity at the join	Smooth
	High quality at waveform level	Vocoded speech (buzzy)
	Less preferred	More understandable
	Best examples are better	Best examples are worse
Footprint	Large run-time data	Small run-time data
Robustness	Hit or miss (with spurious errors, quality is severely degraded)	Stable
Voice modification	Extremely difficult	Flexible to change speaking types voice characteristics or emotion
	Fixed voice	Various voices

synthesis uses both text features (phonetic and prosodic contexts are typically used) and speech features (i.e. spectral and acoustic features) to calculate and minimize the target cost (best units) and concatenation cost (the best sequence) of units for an utterance.

Those techniques have received considerable attention and resources in improving the synthetic voice. Each technique has pros and cons, hence has been adopted in different applications and domains. As a result, the difference between synthetic voices of both approaches and the human voice has been small enough in terms of naturalness for their real-life applications. Unit-selection synthesis tends to be more suitable for applications having limited domain with high quality voice, such as transportation announcement system (e.g. train station, airport) or call centers (for services 24/7). On the other hand, HMM-based speech synthesis can work well in any applications, especially having open domain such as SMS/email/e-newspaper reading systems, question/answering systems or speech translation systems.

According to the review of Zen et al. (2009), in both the Blizzard Challenge in 2005 and 2006, where "common speech databases were provided to participants to build synthetic voices, the results from subjective listening tests revealed that HMM-based synthetic voice was more preferred (through mean opinion scores) and more understandable (through word error rates)". The best examples of unit-selection synthesis are better than those of HMM-based synthesis.

The HMM-based speech synthesis systems need less memory to store the parameters of the model (statistics of acoustic models), hence smaller run-time data, than memorizing the data (multi-templates of speech units) as unit-selection synthesis systems. Therefore, the HMM-based speech synthesis systems can be constructed with a small amount of training data. This leverages the HMM-based approach in supporting multilingual languages, with the fact that only the contextual features to be used depend on each language.

In unit-selection synthesis, when a required sentence happens to need phonetic and

prosodic contexts that are under-represented in a database, the quality of the synthesizer can be severely degraded. Even though this may be a rare event, a single bad join in an utterance can ruin the listeners' flow. It is not possible to guarantee that bad joins and/or inappropriate units will not occur, simply because of the vast number of possible combinations that could occur. Whereas, HMM-based synthesis is more "robust" than unit-selection synthesis, for instance, to noise/fluctuations due to the recording conditions or the lack of some speech units. This is because adaptive training can be viewed as "a general version of several feature-normalization techniques such as cepstral mean/variance normalization, stochastic matching, and bias removal" (Zen et al., 2009).

The main advantage of statistical parametric synthesis including the HMM-based approach is its flexibility in changing its voice characteristics, speaking styles, and emotions. This is still problematic with unit-selection synthesis in spite of its combination of voice-conversion techniques. However, we can easily change voice characteristics, speaking styles, and emotions in statistical parametric synthesis by transforming the model parameters. There have been four major techniques to accomplish this: adaptation, interpolation, eigenvoice, and multiple regression, cf. Zen et al. (2009). Besides, unit-selection synthesis usually requires various control parameters to be manually tuned. Statistical parametric synthesis, on the other hand, has few tuning parameters because all the modeling and synthesis processes are based on mathematically well-defined statistical principles (Zen et al., 2009).

The major disadvantage of statistical parametric synthesis against unit-selection synthesis is the quality of the synthesized speech. The three degrading quality factors are: (i) vocoders (i.e. synthetic speech of a basic HMM-based TTS system sounds buzzy with a mel-cepstral vocoder with simple periodic pulse-train or white-noise excitation), (ii) acoustic modeling accuracy (from which speech parameters are directly generated) and (iii) over-smoothing (i.e. detailed characteristics of speech parameters are removed in the modeling part and cannot be recovered in the synthesis part). Many research groups have contributed various refinements to achieve state-of-the-art performance of HMM-based speech synthesis (cf. Zen et al. (2009) for details).

1.4 Vietnamese language

Vietnamese, the official language of Vietnam, belongs to the Mon-Khmer branch of the Austroasiatic family. The majority of the speakers of Vietnamese are spread over the South East Asia area as well as by some overseas, predominantly in France, Australia, and the United States (Kirby, 2011). The pronunciation of educated speakers from Hanoi, the capital of Vietnam, in general, is the most widely accepted as a sort of standard (Thompson, 1987).

Vietnamese text is written in a variant of the Latin alphabet (chữ quốc ngữ) with additional diacritics for tones, and certain letters. This script has been existing in its current form since the 17th century, and has become the official writing system since the beginning of the 20th (Le et al., 2010). However, there are a number of characteristics of Vietnamese that distinguish it from occidental languages.

First, Vietnamese is a tonal language, in which pitch is mostly used as a part of speech, changing the meaning of a word/syllable. There are six different lexical tones in the writing system, each tone can contribute to the creation of the morpheme and the meaning of a word/syllable, e.g. "ba" *(father)* – level tone, "bà" *(grandmother)* – falling tone, "bã" *(residue)* - broken tone, "bả" *(bait)* – curve tone, "bá" *(aunt)* – rising tone, "bạ" *(strengthen)* – drop tone. The tones make the Vietnamese language have a musical characteristic; make sentences rhythmic and melodious (Nguyen, 2007).

Second, Vietnamese is an "inflectionless language in which its word forms never change". Vietnamese lacks morphological markers of grammatical case, number, gender, tense, and hence it has no finite/non finite distinction. In other words, Vietnamese words do not change depending on grammatical categories, e.g. "bạn" is the same for singular and plural (in contrast to "student" and "students" in English), the same for male and female (in contrast to "ami" and "amie" in French)... This inflectionless characteristic makes a "special linguistic phenomenon common in Vietnamese: type mutation, where a given word form is used in a capacity that is not its typical one (a verb used as a noun, a noun as an adjective...) without any morphological change. For example, the word "yêu" may be a noun (the devil) or a verb (to love) depending on context" (Le et al., 2010).

Third, Vietnamese is a non-affix language in contrast to the means of generating antonyms in English/French by the prefixes "im-", "ir-", "un-", e.g. "impolite", "unreadable", "irregular"... That is to say Vietnamese word structure does not use the affixes (prefixes, suffixes or infixes) (Nguyen, 2007).

And fourth, Vietnamese is an isolating language, the most extreme case of an analytic language, in which the boundary of syllable and morpheme is the same, each morpheme is a single syllable. Each syllable usually has an independent meaning in isolation, and polysyllables can be analyzed as combinations of monosyllables (Doan, 1977). Hence, a syllable in Vietnamese is not only a phonetic unit but also a grammatical unit (Doan, 1999b). Lexical units may be formed of one or several syllables, always remaining separate in writing. Although dictionaries contain a majority of compound words, monosyllabic words actually account for a wide majority of word occurrences. This is in contrast to synthetic languages, like most Western ones, where, although compound words exist, most words are composed of one or several morphemes assembled so as to form a single token (Le et al., 2010). Some examples can be found in different languages are presented in Example 1.

Example 1 Syllables, morphemes and words in English, French and Vietnamese (Nguyen, 2007)

- In English: The word "unladylike" has three morphemes (un)(lady)(like) and four syllables (un)(la)(dy)(like), while the word "dogs" has two morphemes (dog)(s) and one syllable.
- In French: The word "école" (school) has two syllables (é)(cole) and one morpheme, while the word "vendeur" (seller) has two syllables (ven)(deur) and two morphemes (vend-eur).
- In Vietnamese : The sentence "Đẹp vô cùng tổ quốc ta ơi!" *(How beautiful our country is!)* has seven morphemes, seven syllables: (Đẹp)(vô)(cùng)(tổ)(quốc)(ta)(ơi), and five words including three mono words: (đẹp), (ta), (ơi) and two compound words: (vô cùng), (tổ quốc).

Fifth, the Vietnamese language has adopted quite many words from foreign languages, such as tiếng Hán (Chinese) and French. For example, the words "đấu tranh" *(struggle)*, "giai cấp" *(class)*, "nhân nghĩa" *(benevolent and righteous)* are tiếng Hán, while "nhà ga" *(gare)*, "xà phòng" *(savon)*, "cà phê" *(café)* are French (Nguyen, 2007).

And finally, Vietnamese is a "quite fixed order language, with the general word order SVO (subject-verb-object)". As for most languages with relatively restrictive word orders,

Vietnamese relies on the order of constituents to convey important grammatical information (Le et al., 2010).

1.5 Current state of Vietnamese TTS

Vietnamese TTS recently has been receiving more attentions due to its' necessity in real-life applications.

The *Sao Mai Vietnamese reader* (or *'Sao Mai voice'* for short) of the Sao Mai Vocational and Assistive Technology Center for the Blind [8], Ho Chi Minh city is considered the first (2004) and most common software on Windows for the blinds due to its ease of use. This project came from a World Bank prize in the competition of the Vietnam 2003 Innovation Day (Tran, 2007a, 2013). The concatenative synthesis was adopted for the synthesis engine of *'Sao Mai voice'*. Syllables were chosen as speech units. The syllable corpus of this software included about 16.000 syllables (isolately recorded): more than 7000 Vietnamese words and nearly 9000 loanwords.

However, the main disadvantages of the *Sao Mai voice* are: (i) the low quality of synthetic speech (ii) the different voices for different text encodings. Two main reasons for its low quality are (i) the low-quality recording environment of corpus (from 1990s) (ii) the discontinuity at join points between isolately-recorded syllables. Since Vietnamese is a tonal language, the synthetic voice has more discontinuity issues with intonation (e.g. "out-of-tune"). Although the quality of this software is not good, it is still mainly used by the Vietnamese blinds on the platform of Windows until now. The reasons were found as follows: (i) it can be compatible to support any applications on Windows (ii) it can be integrated to JAWS [9], the most popular screen reader (in English) for the Vietnamese blind (iii) it facilitates the Blinds to follow the content of text on screen or applications, such as reading by characters, by syllables (iv) there is currently no better Vietnamese TTS system targeting the blinds (Tran, 2013).

There have been a few other works on Vietnamese speech synthesis using formant or concatenative synthesis techniques (Do and Takara, 2003, 2004) or Nguyen et al. (2004). The Hanoi Vietnamese dialect was chosen for both studies. In the work of Đỗ Trọng Tú (Do and Takara, 2003, 2004), a Vietnamese TTS system (VieTTS) was built as a parametric and rule-based speech synthesis system. Fundamental speech units of this system were half-syllables with the level tone. VieTTS uses a source-filter model for speech production and a Log Magnitude Approximation (LMA) filter as the vocal tract filter. Tone synthesis of Vietnamese was implemented by using F0 patterns and power pattern control. The second work (Nguyen et al., 2004) integrated the Fujisaki model (Hiroya Fujisaki, 1984) into VnVoice, a concatenative Vietnamese TTS system, based on a set of rules to control the F0 contour. In general, the quality of the synthetic voices of VieTTS and VnVoice were acceptable but still had limitations of formant or concatenate synthesis techniques. The work of Nguyen et al. (2004) had a better quality but still met problems in controlling F0 and duration.

Since Vietnamese is a tonal language with a number of lexical tones, we face a great challenge in building a high-quality TTS system. In the following subsections, we will present state-of-the-art work on Vietnamese TTS, using unit selection and HMM-based synthesis techniques.

8. http://www.saomaicenter.org/vi/tts/
9. http://www.freedomscientific.com/Products/Blindness/JAWS

1.5.1 Unit selection Vietnamese TTS

Trần Đỗ Đạt (Tran, 2007b) built a unit-selection TTS system for Hanoi Vietnamese using di-phones and half-syllables as speech units, called *'HoaSung'* (means "water-lily flower"). The corpus of *HoaSung*, called VNSpeechCorpus (hereafter called "VNSP" for short), was collected and filtered by different resources (e.g. stories, books, and web documents) from websites. It included various types of data: words with six lexical tones, figures and numbers, dialog sentences and short paragraphs. It comprised about 630 sentences in 37 minutes, recorded by a TV broadcaster from Hanoi. The CART model was chosen to construct a duration template (Tran et al., 2007), and the Time-Domain PSOLA (TD-PSOLA) algorithm was adopted for manipulating the pitch and timing of speech units. A linguistic feature set built for modeling units duration included: (i) phonetic features, e.g. articulation place/manner, positions of phonemes, (ii) context-based features: e.g. preceding/succeeding phoneme, positions of phoneme in the current syllable. An intonation model was proposed to generate the F0 contour of synthetic utterances. This model was built based on the results of the analysis on relations among factors that influenced the intonation in Vietnamese: (i) the tones that make up the sentence, (ii) the register of each tone, (iii) the influence of tonal coarticulation phenomena and (iv) the duration of the syllable (Tran and Castelli, 2010).

The subjective results of *HoaSung* indicated that the system using half-syllables as speech units gave a better quality than the one using di-phones. HoaSung can be considered a quite complete TTS system with a rather high quality synthetic speech. However, we have found that the phone set of *HoaSung* did not cover the latest phonology system of modern Hanoi Vietnamese, such as the merge of [s] and [ʂ] or the appearance of new phonemes in loanwords. The main limitations reported in the work of (Tran, 2007b) were: (i) the inability to reproduce the important changes in fundamental frequency due to glottalization phenomenon (ii) the small corpus that was not able to synthesize syllables composed of half-syllables not in the corpus (iii) the issues in automatic analysis of text such as text normalization, word segmentation, POS tagger (iv) the lack of F0 modeling for sentence modes.

The work of Le et al. (2011) partly addressed the last problem of *HoaSung* for yes/no questions without auxiliary verbs. Compared to the declarative intonation, in this type of question, the whole F0 contour was raised by a number of percentages of the F0 mean (normalized register ratio) and the contour of the final syllable was raised by a number of percentages of the F0 mean (increasing slope) (Le et al., 2011). This model was applied to *HoaSung* and gave some positive results. However, it did not work well in some particular final syllable tones, e.g. falling tones, curve tones due to the small analysis corpus. Moreover, although the duration relates to the F0 contour, it was not studied and modeled in this work.

HoaSung was extended using the non-uniform unit selection technique (Do et al., 2011) to build the second version. The same speech corpus was used, but annotated at the syllable level with some necessary information such as phonemic elements, tone, duration, energy and other contextual features. The sentences in the text corpus were parsed into syntactic phrases, i.e. phrase-trees. In this work, speech units were not anticipatively determined, but varied according to the availability of the speech corpus. The main idea was to minimize the number of join points, which put a higher priority to longer available speech units. If there were lack of samples as syllables or above syllables (e.g. words, phrases) when searching units, the half-syllable corpus of *HoaSung* was used. The preliminary perceptual result showed an improved quality of synthetic speech of the new system. However, the test corpus was not well designed to cover all instances of combining speech units. Moreover, there was no connection between the process of choosing speech units as syllables or above and the process of choosing units as half-phones in calculating target cost and concatenation cost. As a result, the total

cost was not optimized for utterances needing half-syllables.

'Voice Of Southern Vietnam' (*VOS*) was developed by Vũ Hải Quân and his team at the AILab [10], Ho Chi Minh University of Science. This system was first built using concatenative synthesis with phrases as speech units (Vu and Cao, 2010). The latest version of *VOS* used non-uniform unit selection synthesis with speech units as syllables or above and a very large corpus, a typical speech of the Southern dialect of Vietnam. The quality is better in the limited domain (e.g. football commentaries), which has only a few number of concatenation points. However, transitions between join sections of the waveforms did not sound smooth/-continuous, especially for utterances synthesized by a number of speech units. This system targeted to build a new voice reader for the Vietnamese blind, however it has not been used in real-life due to its usability limitations. To the best of our knowledge, there is no publication related to the latest version of the system.

A few other Vietnamese TTS systems (e.g. *eSpeak* [11], *vietTalk*, *vnVoice*) have been built to support Vietnamese blind people in using personal computers or smart phones. However, most of these systems have been rarely used by blind users because of their drawbacks in quality and usability. Since 2014, *vnSpeak* [12] has become available as a TTS engine for Android platform. This system adopted the unit selection technique and provided a number of supporting functions for users to interact with smart phones. This system has received positive feedback from Vietnamese blind users.

1.5.2 HMM-based Vietnamese TTS

Many works on HMM-based speech synthesis for the tonal languages have been published, not only for the standard synthetic speech but also for the speech with different speaking styles or the expressive speech. For instances, for Mandarin, the work of Duan et al. (2010), Guan et al. (2010), Hsia et al. (2010), Qian et al. (2006), Yu et al. (2013), Zhiwei Shuang (2010) focused on basic problems of improving the naturalness of HMM-based synthetic speech. Mixed-language or bi-lingual speech synthesis was studied in the work of Qian and Soong (2012), Qian et al. (2008) while Li et al. (2010, 2015) worked with expressive speech. The HMM-based speech synthesis for Thai put attention in tone correctness improvement, such as the work of Chomphan (2011), Chomphan and Chompunth (2012), Chomphan and Kobayashi (2007, 2008), Moungsri et al. (2014); or in speaker-denpendent/indenpendent in Chomphan (2009), Chomphan and Kobayashi (2009).

For the Vietnamese HMM-based speech synthesis, to the extent of our knowledge, there are only two following main groups: (i) from the Institute of Information Technology (IoIT, which belongs to the Vietnamese Academy of Science and Technology) (Dinh et al., 2013, Phan et al., 2013a, 2012, 2013b, 2014, Vu et al., 2009) and (ii) from the Yunnan university, China (He et al., 2011, Kui et al., 2011). Both groups followed the core architect of HTS to develop TTS systems for Hanoi Vietnamese.

We assumed that the first publication on Vietnamese HMM-based speech synthesis was the work of IoIT (Vu et al., 2009). This system simply applied the HTS for Vietnamese with the training corpus including 3000 phonetically-rich sentences, semi-automatically labeled at phoneme-level. Hanoi Vietnamese phonetic and phonology and tonal aspects were considered when building the phone and feature sets for the system. Features at phoneme, syllable,

10. http://www.ailab.hcmus.edu.vn/
11. http://espeak.sourceforge.net/. This is an open-source speech synthesizer that Google Translate uses to supports Vietnamese since 2010. Whereas, multilingual well-known TTS systems (cf. Section 1.2) do not support Vietnamese.
12. http://www.vnspeak.com/

word (including Part-Of-Speech POS), phrase and utterance level were chosen. There were additional features of tone types of preceding, current and succeeding syllable, compared to the feature set of English. This work reported that the intelligibility of the synthetic utterances is approximately 100%, and the quality of synthesis speech ranges from fair to good (3.23 on a 5 point MOS scale) through the preliminary evaluations (number of subjects was not mentioned).

It appears that the work of the group from the Yunnan university (He et al., 2011, Kui et al., 2011) also simply adopted the HMM-based synthesis technique for Vietnamese using the STRAIGHT synthesizer [13]. About 600 labeled sentences are used to train the HMM model. A preliminary evaluation (10 subjects) was carried out with the same conclusion on the synthetic voice as the work of Vu et al. (2009). To our knowledge, there were no more studies or experiments for further analysis or improvements.

Several other publications of the first group, IoIT, presented a detail implementation of the HMM-based approach to the Vietnamese TTS with a 400-sentence training corpus Phan et al. (2013a, 2012). The preliminary evaluation only aimed at observing the similarity of spectrogram and pitch contours of natural speech signals and synthetic speech signals. It seems that those publications did not provide any new work, compared to the first (Vu et al., 2009).

Further researches of IoIT Dinh et al. (2013), Phan et al. (2013b, 2014) targeted the same work, which focused on the importance of prosodic features. Additional intonation features adopted from English using the ToBI model were used in these studies, including: (i) phrase-final intonation, and (ii) pitch accent. In the evaluation phase, a MOS test was performed with a natural reference and two TTS voices: (i) without POS and intonation features (ii) with POS and intonation features. Results showed that the voice with prosodic features was about 0.7 higher than the one without those features on 5-point MOS scale. However, there was no further study on the impact of individual POS or intonation features to the synthetic voice.

1.6 Main issues on Vietnamese TTS

The initial motivation of this work was to build a high-quality TTS system assisting Vietnamese blind people to access written text. The scope of this work was then narrowed to build a high-quality TTS system with unlimited vocabulary. Based on all the above analyses on the two state-of-the-art TTS techniques, the HMM-based approach was chosen to build a TTS system for Vietnamese. Beside the predominance on general quality, footprint and robustness; there exists a core part from HTS, and a number of supporting platforms to build an HMM-based TTS system.

This section gives main issues that we encountered during the realization of our TTS system. General solutions of this research for these issues are also introduced.

1.6.1 Building phone and feature sets

In HMM-based speech synthesis, many contextual features (e.g., phone identity, locational features) are used to build context dependent HMMs. However, due to the exponential increase of contextual feature combinations, a decision-tree based context clustering is the most common technique to cluster HMM states and share model parameters among states in each cluster. Each node (except for leaf nodes) in a decision tree has a context related question.

13. http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTadv/index_e.html

Acoustic attributes of phonemes or contextual features are used to build questions, such as *"Is the current phoneme a semivowel?"*, *"Is the previous phoneme voiced?"*. Hence, there is a need build a proper acoustic-phonetic unit set to develop an HMM-based TTS system for a specific language. This unit set is also essential for the automatic labeling of a training corpus, in which it is used to model and to identify clear acoustic events, which an expert phonetician would mark as boundaries in a manual segmentation session.

Due to the automation in HMM clustering, the semantics of the contextual features (e.g. the importance or the weight of these features) may not be well considered. Hence, some crucial features of Vietnamese, such as lexical tones, may do not have proper priorities in building decision tree, which may lessen the impact of these features on improvement of the synthetic speech quality. To our best of knowledge, in other work or for other tonal languages, lexical tones may be explicitly modeled in a TTS system (Shih and Kochanski, 2000)(Do and Takara, 2003)(Tran and Castelli, 2010). In the Thai language, tone correctness of the synthetic speech may be improved by investigating the structures of the decision tree with tone information in the tree-based context clustering process of the HMM-based training (Chomphan and Kobayashi, 2008)(Chomphan and Chompunth, 2012)(Moungsri et al., 2014).

To address above issues for Vietnamese TTS, in this work, we built an acoustic-phonetic unit set in tonal context, in which a new speech unit was proposed for an allophone with respect to lexical tones, called "tonophone". The lexical tones hence might have been "modeled" with a highest priority in the context clustering process of the HMM-based training. Furthermore, previous Vietnamese TTS systems were mostly developed for Hanoi, a standard Vietnamese dialect. However, the phonetic analysis of those works did not cover the latest phonology system of modern Hanoi Vietnamese, such as merging [s] and [ʂ] or appearance of new phonemes in loanwords, such as the initial consonant [p]. In this research, a complete acoustic-phonetic tonophone set was built on the basis of a literature review on the latest phonetics and phonology of modern Hanoi Vietnamese.

1.6.2 Corpus availability and design

Several works on Vietnamese speech corpus were presented, but mainly for speech recognition (Le et al., 2004, 2005, Vu and Schultz, 2009, 2010, Vu et al., 2005). These works did not focus on designing the text corpus, but on collecting/recording the speech corpus, as well as on selecting speakers or on automatic alignment.

In the work of Tran (2007b), a corpus for a unit selection TTS system was collected from different resources from the Internet (e.g. stories, books, web documents), and manually chosen by experts. It included 630 sentences with various types of data: words with six lexical tones, figures and numbers, dialog sentences and short paragraphs. However, due to the small size, this system was not able to synthesize a number of syllables composed of half-syllables not in the corpus. The work of Vu et al. (2009) reported a 3000-sentence training corpus composing phonetically-rich sentences for spoken Vietnamese. Moreover, that corpus is not available for other researchers. Other studies reported several-hundred-sentence corpora without any investigation or design (Tran, 2007b)(Dinh et al., 2013)(Kui et al., 2011).

As a result, to the best of our knowledge, we assumed that there lacks a work on analysis and design of a text corpus for Vietnamese TTS. Since Vietnamese is a tonal language, the training corpus should not only cover phonemic context (e.g. di-phones) but also tonal context. Based on other works on corpus design, we investigated on design phonetically-rich and -balanced corpora for Vietnamese TTS using a huge raw text crawled from various sources. A training corpus for an HMM-based TTS system was designed to cover 100% di-tonophones (i.e. an adjacent pair of "tonophones").

1.6.3 Building a complete TTS system

The work of Tran (2007b) presented a complete architecture of Vietnamese concatenative TTS, which composes of both high-level and low-level speech synthesis. However, there were still numerous issues in automatic analysis of text such as text normalization, word segmentation, or POS tagger. To our best of knowledge, most of previous research on HMM-based Vietnamese TTS (Vu et al., 2009)(Kui et al., 2011)(Dinh et al., 2013)(Phan et al., 2013b) adopted HTS (HMM-based Speech Synthesis System) [14] framework for experiment. They presented only the core architecture from HTS (Zen et al., 2007), which mainly presents training and synthesis parts. All processes in these two parts can be performed using existing tools from HTS or other frameworks. However, the text analysis or the natural language processing part was not investigated in detail.

Although Vietnamese is an alphabetic script, there existed issues in automatic text analysis in the high-level such as text normalization, word segmentation, POS tagger due to a number of the language's distinguishable characteristics from the occidental languages. Spaces and punctuations in the occidental languages can be used as the main predictors of word segmentation, yet in Vietnamese, there is no word delimiter or specific marker that distinguishes the boundaries between words. Blanks are not only used to separate words, but they are also used to separate syllables that make up words. Moreover, the Vietnamese language creates complex words by combining syllables that most of the time possess an individual meaning. As a result, there are ambiguities in word segmentation that need to be addressed. Real texts in Vietnamese often include many Non-Standard Words (NSW) that one cannot find their pronunciation by using "letter-to-sound" rules (e.g. numbers, abbreviations, date). In addition, 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 by context. This raises a real problem in text normalization. Vietnamese is an "inflectionless language in which its word forms never change", regardless of grammatical categories, which leads to a special linguistic phenomenon common in Vietnamese, called "type mutation", where a given word form is used in a capacity that is not its typical one (a verb used as a noun, a noun as an adjective...) without any morphological change" (Le et al., 2010). This property introduces a huge ambiguity in POS tagging.

In this work, we presented a complete architecture of an HMM-based TTS system, composing three parts: natural language processing, training and synthesis part. Constituent modules in the natural language processing part were investigated and constructed. As a result, a complete HMM-based TTS system for Vietnamese was built in this work.

1.6.4 Prosodic phrasing modeling

HMM-based speech synthesis provides a statistical and machine learning approach, in which speech parameters and contextual features are force-aligned to build trained models. Each HMM also has its state-duration distribution to model the temporal structure of speech. As a result, prosodic cues such as intonation, duration can be well learned in context. This considerably increases the naturalness of the synthetic voice. The remaining problem in prosodic analysis is prosodic phrasing, including pause insertion and lower levels of grouping syllables. In an HMM-based TTS system, a pause is considered a phoneme; its duration hence can be modeled. However, the appearance of pauses cannot be predicted by HMMs. Lower phrasing levels above words may not be completely well modeled with basic features.

14. http://hts.sp.nitech.ac.jp/

As aforementioned, the "type mutation" property of Vietnamese introduces a huge ambiguity in Part-Of-Speech (POS) tagging, hence in automatically identifying function or content words. As a result, although function words in occidental languages are good candidates to predict boundaries of prosodic phrasing, they may not be effectively used in automatic TTS. Besides, punctuations cannot be used as an only clue for pauses or breaks when reading Vietnamese text. Both syllables and words in Vietnamese are separated by spaces; hence it is not easy to determine the word boundaries. Vietnamese input text thus is a sequence of syllables, separated by spaces. This leads to a big issue in prosodic phrasing in Vietnamese TTS, which may need higher-level information from text – syntax.

Due to the constraint with the lexical tones, the utterance-level intonation in Vietnamese language might be less important in prosodic phrasing than that in other intonational languages (e.g. English, French). In this research, we aimed at prosodic phrasing for the Vietnamese TTS using durational clues alone.

1.6.5 Perceptual evaluations with respect to lexical tones

We assumed that the lexical tones were important not only in building but also in evaluating TTS systems for Vietnamese. However, most related works carried out the MOS and/or intelligibility test, which were used for any language, to evaluate the quality of Vietnamese TTS systems. There is a lack of an investigation of perceptual evaluations with respect to lexical tones.

In this work, beside some traditional perception tests (e.g. MOS test), tone intelligibility test was designed and performed for evaluating continuous synthetic speech of our TTS system. This test asked subjects to identify the most likely syllable they heard among a group of syllables bearing different tones in an utterance. A tone confusion pattern was also discussed for relations of tones. The intelligibility test was also carried out with a Latin square design, which eliminated the issue of duplicate contents of stimuli. The error rate of the tone level was also investigated.

1.7 Proposition and structure of dissertation

As aforesaid, this work targets to design and implement a high-quality Vietnamese TTS system using HMM-based approach. The major contributions are the following:

- Proposing a new approach in building a tonophone set (i.e. allophones with respect to lexical tones) for Vietnamese TTS, based on the literature review on the Vietnamese phonetics and phonology;
- Designing and recording a new corpus, called VDTS (Vietnamese Di-Tonophone Speech), to cover both phonemic and tonal contexts for Vietnamese TTS systems;
- Designing an entire architecture (including a complete text analysis/natural language processing phase from text normalization, word segmentation, POS tagging, G2P conversion) and a contextual feature set for an HMM-based Vietnamese TTS system;
- Building VTED (Vietnamese TExt-to-speech Development system), an HMM-based Vietnamese TTS system, following the proposed design and the new corpus VDTS;
- Proposing and evaluating a novel prosodic phrasing model using syntactic blocks (with an automatic Vietnamese syntactic parser) to improve the rhythm of the synthetic voice of VTED;

- Designing, carrying out and analyzing various perceptual evaluations of VTED including MOS test, Intelligibility test, Tone intelligibility test, and Pair-wise comparison test.

The rest of this dissertation is organized as follows.

Chapter 2 presents our literature review on phonetics and phonology of the modern Hanoi dialect of Northern Vietnamese (Hanoi Vietnamese). Different opinions on Vietnamese syllable structure are discussed; the final chosen hierarchical structure with elements is induced. The phonology and tone system of modern Hanoi Vietnamese language are then described in this chapter. Four main results essential in building and evaluating a Vietnamese TTS system as well as designing corpus are described: (i) a set of grapheme-to-phoneme rules, (ii) the Vietnamese phone set regarding lexical tones, in which a new speech unit was proposed: a tonophone, (iii) the acoustic-phonetic tonophone set, which provides acoustic attributes for each segment, and (iv) PRO-SYLDIC, an e-dictionary with transcriptions of all pronounceable syllables in the language.

The proposal of corpus design for Vietnamese TTS is shown in Chapter 3, in respect to phonemic and tonal contexts. This corpus was recorded in a controlled well-equipped studio and pre-processed for a TTS system.

In Chapter 4, a novel prosodic phrasing model using syntactic blocks, syntactic links and POS are described. The chapter then describes the evaluation of the pause prediction performance using Precision, Recall and F-score. Due to the importance of syntax to prosodic phrasing, syntax theory, Vietnamese syntax and Vietnamese syntactic parsing are also covered in this chapter (details are described in Appendix A). Automatic syntactic parsing approaches and several parsing types for the adopted Vietnamese syntax parser are also discussed in this appendix. This chapter also gives an introduction to another proposed prosodic phrasing model using syntactic rules, which is presented in detail in Appendix B.

Chapter 5 first gives an introduction to HMM-based speech synthesis as well as its main processes, i.e. parameter modeling, parameter generation, vocoder. A design as well as the phone and feature sets of a complete Vietnamese HMM-based TTS system are then presented. This chapter then describes the implementation of such a system – VTED – under the platform of Mary TTS. Several synthetic voice versions of VTED using different training corpora and/or feature sets for the perceptual evaluations are provided.

Chapter 6 shows our design and implementation of different perception tests on VTED. The perceptual results are then statistically analyzed for each test. Some GUI test screens and examples of test corpus are illustrated in Appendix C. A summary of the work is given in Chapter 7, which depicts several perspectives of this research.

Chapter 2

Hanoi Vietnamese phonetics and phonology: Tonophone approach

Contents

2.1 Introduction

> Vietnamese, the official language of Vietnam, is spoken natively by over seventy-five million people in Vietnam and greater Southeast Asia as well as by some two million overseas, predominantly in France, Australia, and the United States. The genetic affiliation of Vietnamese has been at times the subject of considerable debate (...). Scholars (...) maintained a relation to Chinese, while Maspero (1912), despite noting similarities to Mon-Khmer, argued for an affiliation with Tai. However, at least since the work of Haudricourt (1953), most scholars now agree that Vietnamese and related Vietic [1] languages belong to the Mon-Khmer branch of the Austroasiatic family.
>
> –(Kirby, 2011, p. 381)–

.

Studying the Vietnamese language, especially its phonetics and phonology, is necessary to understand a language as a means of communication between people; hence plays important roles in speech processing. This chapter recapitulates the phonetic and phonology of the modern Hanoi, which is widely considered as the standard language of Vietnamese.

Section 2.2 presents some discussions of different scholar and finally gives our conclusion on Vietnamese syllable structure. The phonological system of Hanoi Vietnamese is described in Section 2.3 while the lexical tones are discussed in Section 2.4. Based on this literature review, main grapheme-to-phoneme rules are provided in Section 2.5. In Section 2.6, tonophone, a new speech unit i.e. a phone concerning a corresponding lexical tone, is introduced. The construction of the Vietnamese tonophone set with acoustic attributes is described. Section 2.7 gives an analysis of Hanoi Vietnamese rhymes and syllable orthographic rules in order to build an e-dictionary with transcriptions of pronounceable syllables. Those results were used for building our TTS system as well as designing corpora.

The convention of phonetic notation in this work is adopted from the study of Laver (1994). In order to distinguish phonetic transcription from orthographic and other symbols, phonetic symbols are enclosed in square brackets, e.g. the orthographic representation "trăng" in Vietnamese (enclosed in double quotes) will be transcribed phonetically as [tɕăŋ] (without a lexical tone) or [tɕăŋ-1] (with a lexical tone – level tone 1). This notation is actually transcribed for phonetic realization of phonemes, i.e allophones – "members of a given phoneme" (Laver, 1994, p. 42), called "allophonic transcriptions". Whereas, the choice of symbols in a phonemic transcription, enclosed in slant brackets, is limited to one symbol per phoneme (e.g. /a/) (Laver, 1994, p. 550). To give a better illustration for examples, English meanings of Vietnamese syllables/words are provided and enclosed in round brackets and in *italic* format, e.g. in "trăng" *(moon)*. Phonemes, allophones, phones and are represented using symbols of the International Phonetic Alphabet (IPA).

2.2 Vietnamese syllable structure

The analysis of syllable structure has a direct bearing on the analysis of the phonemic system: numerous nucleus/vowels, combination of glide and vowel, and also central for a tone system.

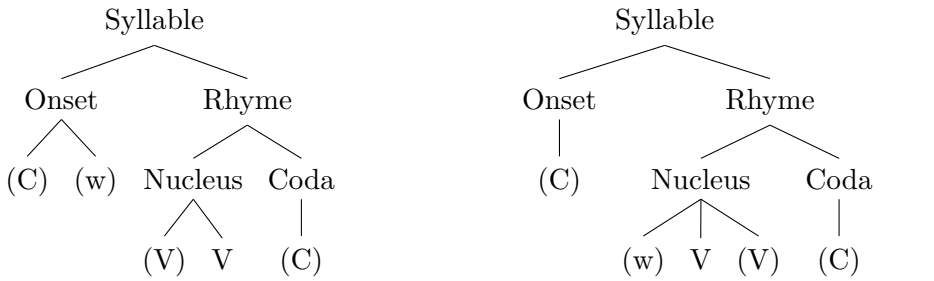
1. The Vietic branch is sometimes referred to as Việt-Mường, although this latter term is also used to refer exclusively to a sub-branch of Vietic containing Vietnamese and Mường.

In this section, several discussions on Vietnamese syllable structure are provided, and the concluded structure for Vietnamese syllables is presented.

2.2.1 Syllable structure

In Vietnamese, as presented in the previous section, the boundary between a phonetic unit (syllable) and a grammatical unit (morpheme) is the same. This characteristic cannot be found in inflectional languages, e.g. Indo-European languages. In addition, each syllable in Vietnamese has a stable and complete structure composed of perceptually distinct units of sound, i.e. phoneme. As a result, the role of syllables in Vietnamese is much different from that in Indo-European languages.

Syllable structure in Vietnamese permits "only single consonants in onset and coda positions, and a single vowel or a diphthong in the nucleus" (Vogel et al., 2004). Most researchers preferred the hierarchical structure of Vietnamese syllables (Thompson, 1987)(Doan, 1977)(Vogel et al., 2004)(Doan, 1999a), however there are different ideas on composite parts and their relationship. Topical issues in this section include (i) the appearance of "medial" part, (ii) the presence of rhyme, (iii) the role of lexical tones in Vietnamese syllable structure.



(a) /w/ in onset by Thompson (1987) (b) /w/ in nucleus by Vogel et al. (2004)

Figure 2.1 – The position of "medial" /w/ in Vietnamese syllables: (a) Thompson (1987) and (b) Vogel et al. (2004)

Medial. There is one apparent complication the so-called "medial", i.e. a glide, a /w/ that may appear between an onset and a nucleus. Scholars such as Vogel et al. (2004) provided an analysis on the basis of a moraic approach for Vietnamese syllables to argue that the glide /w/ is preferred to be in nucleus, not in onset – in contrast with the work of Thompson (1987), illustrated in Figure 2.1. Moreover, Vogel reinforced the proposal by giving some examples of a type of a word game: "Nói lái", which "exchanges different parts of syllables in word sequence to form a sort of spoonerism". On the basis of possibilities manifested in this game, the "medial" is always actually exchanged along with the nucleus, not the initial (Example 2). Vogel's arguments, however, are strong supports for a claim that the "medial" is not part of the onset, but not enough to affirm that it belongs to the nucleus or another crucial part in the hierarchical structure of Vogel – **rhyme**. A good reason to analyze the medial /w/ as a part of rhyme is that it can appear in onset-less syllables in Vietnamese, such as "oan" [wan] *being victim of a glaring injustice*, or "uyên" [wiən] *in "uyên bác" – erudite*.

Đoàn Xuân Kiên (Doan, 1999a) also considered that the glide /w/ is not in onset, but controverted the existence of "medial" in the structure. The scholar argued that the medial should be considered a "semi-vowel" instead of a part of the structure, and adopted the hierarchical structure of Vietnamese syllable, without rhyme. Đoàn Xuân Kiên concluded that there is no persuasive argument on phonetics and phonology for the big role of rhyme

Example 2 The game "Nói lái" with the origin word: "Tuyên bố" [twien ɓo] (Vogel et al., 2004)

- Switching the onset node: [t͟wien ɓ͟o] ⇒ [ɓ͟wien t͟o]
- Switching the rhyme node: [t͟wien͟ ɓ͟o] ⇒ [t͟o ɓ͟wien͟]

on the structure of Vietnamese syllables and that exist four parts in this structure: initial, nucleus, ending, and tone (no rhyme). However, rhyme does exist in the hierarchical structure of Vogel et al. (2004), which, as aforesaid, plays an important role in the "Nói lái" game *(switching elements of syllables)*.

Tone			
Initial	Rhyme		
	Medial	Nucleus	Ending

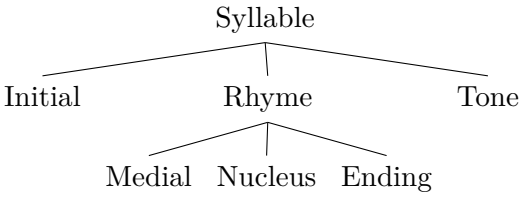


Figure 2.2 – The hierarchical structure of Vietnamese syllables by Doan (1977)

"

Rhyme. Đoàn Thiện Thuật (Doan, 1977) presented main parts in the structure of Vietnamese syllables illustrated in Figure 2.2. The scholar also preferred the hierarchical structure and affirmed the importance of rhyme in the structure of Vietnamese syllables with some analyses on the basis of a moraic approach to the syllables or some word games such as "nói lái", "-iếc hoá", "láy" or "gieo vần", illustrated in Example 3.

In the game "-iếc hóa", a variant of a word in oral conversation is created by adding a new syllable composed of the initial consonant of the original syllable and the rhyme /iek/, e.g. origin syllable: "toán" [twan] ⇒ new word: "toán tiếc" [*t*wan *t*i͟ek͟] *(math)*. "Láy" (reduplication) is the process of creating a new word (called "từ láy" - reduplicated word) by repeating either a whole syllable or part of a syllable. Reduplication in Vietnamese can be applied on initial consonants, e.g. "nhạt nhẽo" [ɲ͟at ɲ͟ɛw] *(insipid)*; rhymes, e.g. "lung tung" [l͟uŋ͟ t͟uŋ͟] *(in utter disorder)*; or both of them, e.g. "tẻo teo", [t͟ɛw t͟ɛw] *(tiny)*. In Vietnamese poems, it is common to find repeated rhymes (strictly following prosody) of verses' syllables, i.e. "gieo vần".

The analysis of "Đoàn Thiện Thuật" mentioned the equivocal characteristic of medial, which raises the ambiguity that the medial is a part of initial or rhyme. However, the number of instances proving that the medial belonging to the initial are uncommon. For instance, in "-iếc hoá", "toán tu*yếc*" [t͟wan t͟w*iek*] *(math)*, in which the repeated parts are initial+medial /tw/, is much less popular than "toán t*iếc*" [t͟wan t͟*iek*], in which only the initial /t/ is repeated. The reduplicated word "lẩn quẩn" [l͟ɤ̆n kw͟ɤ̆n] *(hover about)*, in which only the nucleus and the rarfinal consonant – not the whole rhyme are the repeated parts, is less popular than "luẩn quẩn" [l͟wɤ̆n kw͟ɤ̆n], in which the whole rhyme iterates. In "gieo vần", "qua" [kw͟a] and "mà" [m͟a], in which the repeated part is only nucleus – not the whole rhyme, is a rare example. Based on those analyses, the medial is finally concluded that it is a part of rhyme.

Lexical tones. The last, but important and typical issues of the Vietnamese syllables, are

Example 3 The game "-iếc hoá", "láy", "gieo vần"

- "-iếc hoá": Rhyme is replaced by "-iếc" /iek/ to make a variant of a word in oral conversations, e.g. "toán" [twan] ⇒ variant of word: "toán tiếc" [*t*wan *t*iek] *(math)*; "khoan" [xwan] ⇒ variant of word: "khoan khiếc" [*x*wan *x*iek] *(to drill)*
- "láy": Reduplication of a whole syllable or part of a syllable
  - Reduplication of the initial, e.g. "nhạt nhẽo" [ɲat ɲɛw] *(insipid)*, "mênh mông" [meŋ moŋ] *(spacious)*.
  - Reduplication of the rhyme, e.g. "lung tung" [luŋ tuŋ] *(in utter disorder)*, loắt choắt [lwăt tɕwăt] *(little)*.
  - Reduplication of both initials and rhymes (with or without tone), e.g. "tẻo teo", [tɛw tɛw] *(tiny)*, xinh xinh [siɲ siɲ] *(cute)*.
- "gieo vần": Repeating rhymes of syllables of verses in a poem (strictly following prosody):

  Ao thu lạnh lẽo nước trong veo ([vɛw])
  Một chiếc thuyền câu bé tẻo teo ([tɛw])
  Sóng biếc theo làn hơi gợn tí
  Lá vàng trước gió sẽ đưa vèo ([vɛw])
  (The poem "Thu điếu" of the author Nguyễn Khuyến).

lexical tones and their interaction with other parts in the structure. Some researchers (Le, 1948)(Emeneau, 1951)(Vogel et al., 2004) either did not mention or did not consider Vietnamese tones to be a constituent of syllable structure, since they are not segments and hence cannot be treated as phonemes. However, most scholars such as Doan (1977), Doan (1999a) emphasized the role of lexical tones in the Vietnamese syllable structure. Tones were then treated on a different level from that of segments. In Vietnamese, tones, a mandatory part of syllables, are crucial factors for distinguishing syllables. For instance, "ba" *(father)* – level tone, "bà" *(grandmother)* – falling tone, "bã" *(residue)* - broken tone, "bả" *(bait)* – curve ton, "bá" *(aunt)* – rising tone and "bạ" *(strengthen)* – drop tone are distinct syllables with different meanings.

There are two opinions of the role of tones in a syllable structure: (i) a tone is a prosodic feature, i.e. bringing melody of syllables, not a component part of syllables. (ii) a tone is a non-linear part of a syllable, with other linear parts. The first one is a typical characteristic of polysyllabic languages, where all phonemes are sequentially combined. The melody of syllables can be changed based on contexts. However, a Vietnamese syllable can only bring one stable tone, which makes it different from other syllables. The contribution of each tone could construct the morpheme and meaning of syllable. It is not advisable to arbitrarily modify the tone of a syllable, which may lead to its destruction or falsification.

We concluded that for the relationship amongst the main parts of the structure, tones are non-linear or suprasegmental, i.e. covering and adhering to the whole or a part of syllable, while other parts of syllable are "linear" or segmental, i.e. continuously sequenced distinct segments (Doan, 1999a). Tones appear simultaneously with segmental phonemes to construct a complete structure of syllables. Tones in Vietnamese syllable structure play a typical and

distinguishable role to express perfectly a fully-constituted entity from intonational languages e.g. Indo-European languages.

Đoàn Xuân Kiên discussed about the impact of the tone on the nucleus or on the syllable. Some scholars such as Le (1948) believed that Vietnamese tones mostly adhere to the nucleus, meanwhile others to the whole syllable (Cao, 1975)(Doan, 1977). Trần Đỗ Đạt and his team (Tran et al., 2005) did a perception test using Diagnosis Rhyme Test (DRT) method to discover the effect of the tone on the Vietnamese syllables. The study affirmed that the initial consonant does not carry the information of the tone, and does not take part in the construction the tone of the syllable. As a result, the impact of the Vietnamese tones was concluded only on the rhyme of syllables.

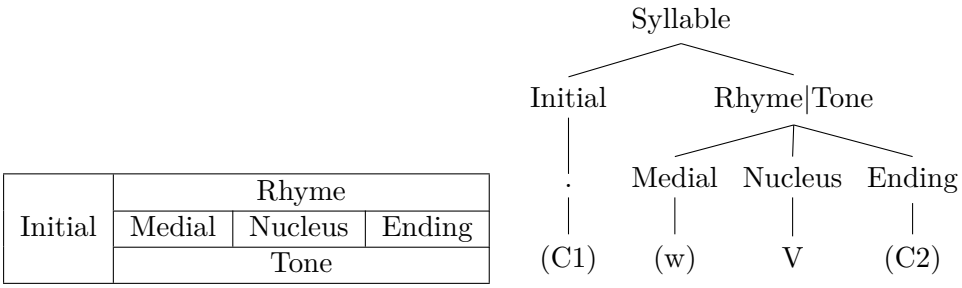


Figure 2.3 – The concluded hierarchical structure of Vietnamese syllables.

From all analyses of previous researches on the structure of Vietnamese syllables, we present the hierarchical structure as shown in Figure 2.3. There are two main parts in a syllable: an initial consonant and a rhyme. A tone appears simultaneously with three segmental elements of rhyme, i.e. medial, nucleus and ending. The nucleus and tone are compulsory while others are optional. As a result, the syllabic structure is (C1)(w)V(C2)+T, where C1 is an initial consonant, w is the semi-vowel /w/, V is a vowel or a diphthong, C2 is a final consonant or a semi-vowel /w j/, and T is a tone (1-4, 5a, 5b, 6a, 6b).

2.2.2 Syllable types

Based on the concluded syllable structure, it can be said that there are 8 structure-based types of Vietnamese syllables, illustrated in Table 2.1. The nucleus is mandatory, hence combined with other optional elements to form 8 groups: (i) nucleus alone (ii) initial+nucleus (iii) medial+nucleus (iv) nucleus+ending (v) initial+medial+nucleus (vi) initial+nucleus+ending (vii) medial+nucleus+ending (viii) initial+medial+nucleus+ending.

Table 2.1 – Structure-based types of Vietnamese syllables

INITIAL	**RHYME**			**Examples**
	Medial	Nucleus	Ending	
		v		“a” /a-1/ *(ah)*, “ử” /ɤ-3/ *(keep warm)*
v		v		“lá” /la-5a/ *(leaf)*, “chờ” /tɕɤ-2/ *(wait)*
	v	v		“oẹ” /we-6a/ *(retch)*, “uy” /wi-1/ *(prestige)*
		v	v	“ích” /ik-6b/ *(helpful)*, “ấy” /ɤ̆-5a/ *(this)*
v	v	v		“loé” /lwe-6a/ *(flash)*, “quá” /kwa-5a/ *(over)*
v		v	v	“trẹo” /tɕew-6a/ *(sprain)*, “nhãn” /ɲan-4 *(longan)*
v	v	v	v	“nhuyễn” /ɲwien-4/ *(fine)*, “soát” /swat-5b/ *(check)*

2.3 Vietnamese phonological system

The Vietnamese phonology has been the subject of strong debates and has drawn the attention of many researchers (Doan, 1977)(Thompson, 1987)(Nguyen and Edmondson, 1998)(Doan, 1999a)(Michaud, 2004)(Michaud et al., 2006)(Haudricourt, 2010)(Kirby, 2011). This work gives a review of the phonological system for the modern Hanoi Vietnamese.

2.3.1 Initial consonants

Table 2.2 presents our adoption with 19 initial consonants for the modern Hanoi Vietnamese. According to Kirby (2011, p. 382), although some previous treatments such as Thompson (1987) recognized an unaspirated, unaffricated palatal stop /c/, in the speech of many younger Vietnamese native speakers from Hanoi, this segment is consistently realized as an affricate [tɕ]. In the initial position, during the production of both the palatal nasal [ɲ] and the palatal affricate [tɕ] are produced, the "tongue body contacts the alveolar or post-alveolar region".

Numerous instances for initial consonants in the modern Hanoi Vietnamese are presented in Example 7. In this dialect, the phonemes "ch-" /c/ and "tr-" /ʈ/ (the first item in Example 7) are "pronounced alike" (Thompson, 1987) and "completely merged in modern Hanoi Vietnamese" to [tɕ] although some varieties of Vietnamese maintain a distinction in the phonetic realizations of orthographic "ch-" and "tr-" (Kirby, 2011). This habit was also taken with two other sets of phonemes: "x-" /s/ and "s-" /ʂ/ are merged to [s] (the second item in Example 7), and "d-" /z/, "gi-" /ʐ/ and "r-" /r/ are pronounced the same as [z] (the third item in Example 7). Examples in the four last items illustrate the rest of the initial consonants in Hanoi Vietnamese.

Table 2.2 – Hanoi Vietnamese initial consonants

Manner of articulation / Place of articulation	Labial		Coronal		Dorsal		Glottal
	Bi-labial	Labio-dental	Dental	Alveolar	Palatal	Velar	
Stop/Plosive	p ɓ		t t^h	ɗ		k	
Nasal	m		n		ɲ	ŋ	
Affricative					tɕ		
Fricative		f v		s z		x ɣ	h
Lateral-approximant			l				

In a smaller number of loanwords (mainly French or proper names from a number of languages), /p r ʂ/ occur, e.g. "pê-đan" [pe ɗan] *(pédale in French)*, "ga-ra" [ga ra] *(garage in French)*. Hence, the /p/ is then adopted as a new phoneme in our work. However, the /r/ is mostly realized as [z] although some speakers, especially young ones who can speak foreign languages, maintain [r] for those loanwords. As a result, the /r/ is not in the initial consonant set of our TTS system.

2.3.2 Final consonants

Hanoi Vietnamese allows eight phonemes in the final position: three unreleased voiceless obstruents /p t k/, three nasals /m n ŋ/, and two approximants /j w/ [2] (Kirby, 2011). In the final position, /t n/ are "canonically alveolar".

2. "Whether these segments are transcribed as final approximants /j w/ or as semivowels is largely a matter of analytic perspective. From a phonological standpoint, these segments may be regarded as approximants

Example 4 Initial consonants in the modern Hanoi Vietnamese

- "ch" and "tr": "cha" [tɕa] *(father)* "tra" [tɕa] *(look up)*
- "x" and "s": "xa" [sa] *(far)* "sa" [sa] *(fall)*
- "d", "gi" and "r": "da" [za] *(leather)* "gia" [za] *(increase)* "ra" [za] *(out)*
- "bi" [ɓi] *(marble)* "mi" [mi] *(eyelashes)* "fi" [fi] *(gallop)* "vi" [vi] *(tiny)*
- "ti" [ti] *(breast)* "thi" [tʰi] *(compete)* "ni" [ni] *(this)* "ly" [li] *(glass)*
- "đi" [ɗi] *(go)* "nhi" [ɲi] *(pioneer)* "nghi" [ŋi] *(doubt)* "ky" [ki] *(stingy)*
- "khi" [xi] *(when)* "ghi" [ɣi] *(write)* "hy" [hi] *(in "hy hữu" - seldom)*

Table 2.3 – Hanoi Vietnamese final consonants

Manner of articulation / Place of articulation	Labial		Coronal		Dorsal		Glottal
	Bi-labial	Labio-dental	Dental	Alveolar	Palatal	Velar	
Stop	p			t		k	
Nasal	m			n		ŋ	
Affricative							
Approximant		w			j		

Some final consonants have variations in phonetic realization, as described in Kirby (2011, p. 383). Although the stops /ŋ k/ following /i e ĕ/ have sometimes been phonetically realized as palatal [ɲ c], they are actually pre-velar [ŋ̟] and [k̟], with no point of alveolar contact. Following back rounded vowels /u o ɔ/, the velar stops /k ŋ/ are produced as doubly articulated labial-velars [k͡p ŋ͡m].

In our work, the orthographies "-anh, -ách" are transcribed as [ĕŋ̟], /ĕk̟/ since the vowels are shortened, from /ɛ/ to [ĕ]. Hence, the velar stops /ŋ k/ are realized as pre-velar [ŋ̟], [k̟] if they follows /i e ĕ/. There do exist a few instances of true velars following /ɛ/, e.g. "xẻng" [sɛŋ] *(shovel)* (Kirby, 2011).

Example 5 Final consonants in the modern Hanoi Vietnamese

- "chích" [tɕiḵ] *(inject)* "trách" [tɕĕḵ] *(blame)* "chiếc" [tɕiək] *(a unit of)*
- "chậc" [tɕɤ̆k] *(well)* "chốc" [tɕuk͡p] *(instant)* "chớp" [tɕɤp] *(flash)*
- "chanh" [tɕĕŋ̟] *(lemon)* "chênh" [tɕĕŋ̟] *(tilted)* "trăng" [tɕăŋ] *(moon)*
- "trang" [tɕaŋ] *(page)* "chung" [tɕuŋ͡m] *(common)* "chum" [tɕum] *(jar)*
- "chan" [tɕan] *(souse)* "chao" [tɕaw] *(swing)* "chai" [tɕaj] *(bottle)*

For a better illustration of those final consonants with several variations, some instances

(consonants) on the grounds that they may not be followed by another consonant. However, these segments are articulated somewhat differently from the initial approximants, with a lesser degree of closure" (Kirby, 2011).

can be found in Example 5. The finals of "chích", "trách", "chanh", "chênh" in the first and the third items of this example are realized as pre-velar: [tɕiḵ], [tɕɛ̆ḵ], [tɕɛ̆ŋ̟], [tɕɛ̆ŋ̟]. Whereas, the phonetic realization of the finals of "chốc", "chung" in the second and the forth item are labial-velars: [tɕu͡kp], [tɕuŋ͡m]. Examples in other items have the standard realization as in the Table 2.3.

2.3.3 Medials or Pre-tonal sounds

A medial is the sound between the initial sound and the nucleus. This element is a lingual and semi-vowel segment, which impacts the timbre of syllables but has no syllabic characteristic (Doan, 1977). Vietnamese has only one medial, transcribed as /w/. This has the same structure as the vowel /u/, but does not get as a nucleus in syllables. For instance, in "chót" [tɕɔt] *(final)*, [ɔ] is the nucleus of the syllable [tɕɔt]. Meanwhile, in the syllables "choắt" [tɕwăt] *(small)* or "chắt" [tɕăt] *(decant)*, [ă] is the nucleus and /w/ is the medial of the syllable [tɕwăt]. The nucleus of a syllable is mandatory and brings the lexical tone in the orthography ("-ó-", "-ắ-"), while the medial is optional and right before the nucleus that carries a lexical tone, hence also called "pre-tonal" sounds.

The medial /w/ increases the volume of the syllable and also contributes to the tone of the syllable (Nguyen, 2007). With a medial glide /w/, a rhyme is produced with the rounding of the nucleus. This rounding is transcribed as a superscript [ʷ] (Michaud et al., 2015). The medial sound hence never precedes the rounded vowels /u o ɔ ŏ/, and never follows the labial consonants /b m f/ except for loanwords. For instance, in "mua" [muə] *buy*, the nucleus is a diphthong /uə/ and there is no medial sound; meanwhile in "qua" [kʷa] *pass away*, there exists a medial sound /ʷ/ and a vowel /a/ as a nucleus.

The orthography of a medial sound is "-u-" if it follows the initial consonant /k/, e.g. "quê" [kʷe] *(hometown)*, "quay" [kʷĕj] *(whirl)*, "quyền" [kʷiən] *(power)* or precedes the close or close-mid vowels, that is /i iə e/, such as "tuy" [tʷi] *(however)*, "tuyến" [tʷiən] *(line)*, "huề" [hʷe] *(draw)*. Its orthography is "-o-" if it precedes the open or open-mid vowels, which are /a ă ɛ ĕ/, such as "hoa" [hʷa] *(flower)*, "xoăn" [sʷăn] *(curly)*, "hoe" [hʷɛ] *(reddish)*, "oanh" [ʷĕŋ̟] *(oriole)*.

2.3.4 Vowels and diphthongs

The nucleus is the main and compulsory sound of the syllable. In Vietnamese, the nucleus is expressed by vowels or diphthongs. Hanoi Vietnamese distinguishes nine long vowels /i e ɛ a ɯ ɤ u o ɔ/, four short vowels /ĕ ă ɤ̆ ŏ/ (Doan, 1977) and three falling diphthongs /iə ɯə uə/ (Kirby, 2011). Diphthongs have the same function as vowels in the syllable (Doan, 1977)(Nguyen, 2007).

Table 2.4 – Hanoi Vietnamese vowels and diphthongs

Elevation of the tongue / **Position of the tongue**	**Front**	**Central**	**Back**	
			Unrounded	Rounded
Close (High vowel)	i	iə ɯə uə	ɯ	u
Close-mid	e		ɤ ɤ̆	o
Open-mid	ɛ ĕ			ɔ ŏ
Open (Low vowel)	a ă			

Table 2.4 illustrates the attributes of Vietnamese vowels and diphthongs. All vowels are

unrounded except for the four back rounded vowels: /u, o, ɔ, ɔ̆/. The vowel with the orthographic "ư" is considered to be close back unrounded /ɯ/ (Doan, 1977, Thompson, 1987) although other researchers might indicate that it is more central than back (Brunelle, 2003, Han, 1966).

Four short vowels /ɛ̆ ă ɤ̆ ɔ̆/ together with their corresponding "long" vowels /ɛ a ɤ ɔ/ reflect their "interpretation as a vowel pair distinguished by phonemic length" in spite of their small spectral differences. It has "not been established that these differences are perceptually or psychoacoustically salient", hence they are transcribed as instances of the same vowel quality (Kirby, 2011). It is economical to use a diacritic for the four short vowels and leave nine long vowels unmarked (Michaud et al., 2015).

The two obvious short vowels /ă ɤ̆/ can be easily found from the orthography "ă â" respectively. The /ɤ̆/ is a close vowel, while the /ă/ is an open one. One of the special cases of the orthography "a" is that it can be transcribed to /ă/ in syllables with "-ay -au" rhymes. Another notable feature of the system is the presence of two other short vowels /ɛ̆/ and /ɔ̆/ corresponding to the orthography "a" in the rhymes "-anh, -ách" and "o" in the rhymes "-ong, -óc". An instance of long-short vowel pairs is "xẻng" [sɛŋ] *(shovel)* and "sảnh" [sɛ̆n̪] *(hall)*; or xoong [soŋ] *(sauce pan)* and "xong" [sɔŋm͡] *(finish)*.

The three diphthongs /iə ɯə uə/ are actually centralizing (Kirby, 2011, Michaud et al., 2015), which brings out the coherence of the system better, as illustrated in Figure 2.4. Doan (1977) and Haudricourt (2010) considered them as front /ie/ or back /ɯɤ uo/.

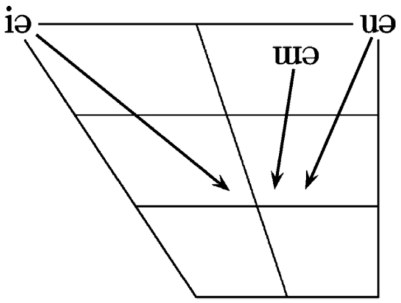


Figure 2.4 – Location of Vietnamese diphthong centroids (Kirby, 2011).

.

Many instances of vowels were also presented in the previous examples. In Example 6, the two first items illustrated three diphthongs with (the first line) or without (the second line) finals. Some exceptional cases of vowels are presented in the last two items. The orthographic "-a-" normally is transcribed as /a/ such as "rang" [zaŋ], "gian" [zan]. However, "-a-" following by "-nh", or "-ch" is pronounced as [ɛ̆], e.g. "ranh" [zɛ̆n̪], "rách" [zɛ̆k̟]. Similarly, "-a-" following by "-u", or "-y" is pronounced as [ă], e.g. "rau" [ză̯w] *(vegetables)*, "ray" [ză̯j] *(rail)*.

Example 6 Vowels and diphthongs in the modern Hanoi Vietnamese

- **Diphthongs:** "tuyết" [tʷiət] *(snow)* "tuốt" [tuət] *(pluck off)* "tướt" [tɯət] *(long)*
- **Diphthongs:** "tia" [tiə] *(ray)* "tua" [tuə] *(fringe)* "tưa" [tɯə] *(fur)*
- **Vowels:** "rang" [zaŋ] *(roast)* "ranh" [zɛ̆n̪] *(mischievous)* "rách" [zɛ̆k̟] *torn*
- **Vowels:** "gian" [zan] *(disloyal)* "rau" [ză̯w] *(vegetables)* "ray" [ză̯j] *(rail)*

2.4 Vietnamese lexical tones

2.4.1 Tone system

> The Vietnamese tone system belongs to the pitch-plus-voice quality type, i.e. the tone is not defined solely in terms of pitch: it is a complex bundle of pitch contour and voice quality characteristics (...). The length of the vowel, and the presence or absence of a final nasal, have no influence on which tones the syllable can bear: there is no need to distinguish "heavy" syllables and "light" syllables, or to posit a division of the rhyme into morae. In contrast to the tone systems of African languages (e.g., typically, Niger-Congo family), in Vietnamese, as in many Asian languages, there are neither tone spreading and floating tones nor downstep. Unlike in some varieties of Chinese (...), there is no tone sandhi in Vietnamese.
>
> –Michaud (2004, p. 121)–

As presented in the Section 2.2, each syllable carries one lexical tone, mainly impacting all elements of its rhyme. In orthography, Hanoi Vietnamese, has six different lexical tones, adhering to the nucleus element in writing scripts.

In our work, due to previous studies and ease of transcription, we adopted a numeric way for representing the tones: (**1**) level tone, e.g. "ta" [ta-1] *(we)*; (**2**) falling tone, e.g. "tà" [ta-2] *(declining)*; (**3**) curve tone, e.g. "tả" [ta-3] *(describe)*, (**4**) broken tone, e.g. "tã" [ta-4] *(diaper)*; (5) rising tone, e.g. "tá" [ta-5] *(dozen)*; and (6) drop tone, e.g. "tạ" [ta-6] *(quintal)*. In spite of the six lexical tones in the writing system, Vietnamese actually distinguishes eight tones in phonetic realization: (i) six tones for sonorant-final syllables, and (ii) two tones for obstruent-final syllables, whose final consonants end in an unreleased oral stop /p t k/. "The historical developments that led to the complex tone system of present-day Vietnamese are by now a textbook example of tonogenesis, the various stages of the process" (Michaud, 2004, p. 121). The obstruent-final syllables may carry either of two tones: rising or drop tones. For ease of comparison, the rising tone in sonorant-final syllables, i.e. syllables not ending in /p t k/, is represented as "**5a**", while the one in obstruent-final syllables are represented as "**5b**". This convention is similar to the drop tone: "**6a**" for the drop tone in sonorant-final syllables and "**6b**" for the drop tone in obstruent-final syllables.

Michaud (2004) adopted the representation with four categories: tones A, B and C for three distinctive tones for sonorant-ending syllables, and obstruent-final syllables constituted a fourth set of syllables, without distinctive tone: category D (see Table 2.5). "At a later stage, a second tonal split (bipartition) involving the disappearance of the opposition between the voiced and unvoiced initial consonants created the current paradigm of six tones for syllables with final sonorants (tones A1 through C2) and two tones for syllables with final obstruents (tones D1 and D2; more strictly speaking, two architonemes)".

The tone system with two types of naming convention of Hanoi Vietnamese can be summarized in Table 2.5. Some constraints for tones and other elements of syllables are described as follows (Ferlus, 2001, p. 298).

- Within syllables ending in vowels, all of the six tones can occur.
- Within syllables in nasal finals (-m -n -nh/-ng) and ancient lateral final only tones derived from level (1), falling (2) tones, rising (5a) and drop (6a) can occur in genuine

Table 2.5 – Hanoi Vietnamese tones (Ferlus, 2001, p. 298)

Initial consonants / **Final consonants**	**Voiced finals**			**Voiceless finals**
Voiceless initials	level *(1 or A1)*	curve *(3 or C1)*	rising *(5a or B1)*	rising *(5b or D1)*
Voiced initials	falling *(2 or A2)*	broken *(4 or C2)*	drop *(6a or B2)*	drop *(6b or D2)*

Vietic words. Tones corresponding to curve (3) and broken (4) tones only exist in borrowed words from Chinese, or in words of expressive origin.

- The curve (3) and broken (4) tones are issued on syllables that are either vowel-final or with the ancient final fricative.
- The tones in syllables with final plosives (-p -t -ch/-c) are realized with the same contour as the rising (5a) and drop (6a) tones, but they constitute a subsystem that contrasts, as a whole, with the subsystem in voiced final syllables.

2.4.2 Phonetics and phonology of tone

A schematic representation of the eight tone templates of Hanoi Vietnamese, based on data from one speaker of Michaud (2004), is illustrated in Figure 2.5. The widely cited description by Doan (1977), Kirby (2011), Michaud (2004), Michaud et al. (2006), Nguyen and Edmondson (1998), Thompson (1987) gives the following account, which is also summarized in Table 2.6.

The terms "glottal stop", "glottal constriction", "creaky voice/laryngealization" and "glottalization", used below to describe voice qualities of Vietnamese tones, are adopted and characterized phonetically from the work of Michaud (2004, p. 120).

> - *Glottal stop* is a gesture of closure that has limited coarticulatory effects on the voice quality of the surrounding segments,
> - *Glottal constriction* (also referred to here as glottal interrupt) is a tense gesture of adduction of the vocal folds that extends over the whole of a syllable rhyme,
> - *Laryngealization* (i.e. lapse into creaky voice), resulting in irregular vocal fold vibration, is not tense in itself,
> - *Glottalization* is used as a cover term for laryngealization and glottal constriction.
>
> –Michaud (2004, p. 120)–

Tone 1 (A1), level tone ("ngang"), is symbolized in the writing system by the absence of any tone mark, e.g. "ba" [ɓa-1] *(three)*, "khuya" [xwie-1] *(late)*. Its contour is "nearly level in non-final syllables not accompanied by heavy stress, although even in these cases it probably trails downward slightly. It starts just slightly higher than the mid point of the normal speaking voice range" (Thompson, 1987). This tone "today in the capital is not often lax but usually modal in voice quality" (Nguyen and Edmondson, 1998).

Tone 2 (A2), falling tone ("huyền"), is represented by the grave accent (`), e.g. "bà" [ɓa-2] *(grandmother)*, "tuần" [twɤ̆n-2] *(week)*. It "starts quite low and trails downward towards the bottom of the voice range. This tone is lax and often accompanied by a kind of breathy voicing (voiceless + modal), reminiscent of a sigh" (Thompson, 1987). For some speakers, the tone 2 is "even lax to the point of breathiness with somewhat lowered sub-glottal air pressure" (Nguyen and Edmondson, 1998).

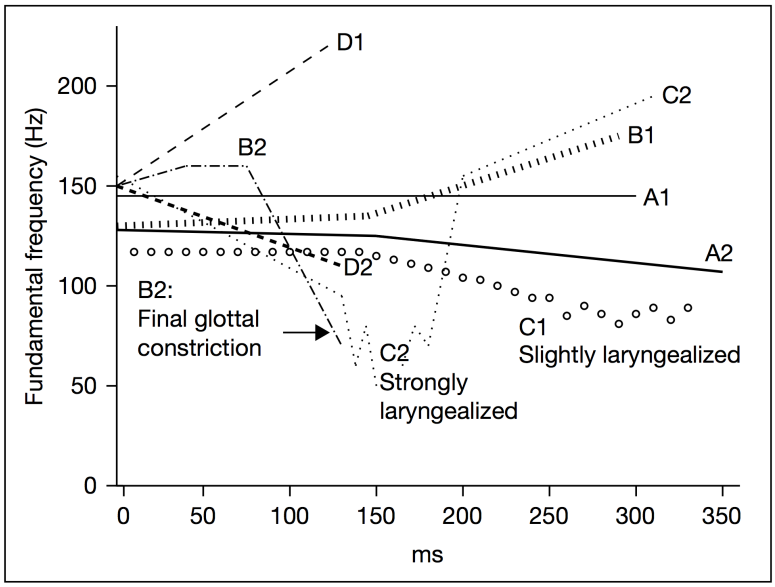


Figure 2.5 – Eight tone templates of Vietnamese tones (Michaud, 2004): A1 (level tone 1), A2 (falling tone 2), C2 (broken tone 3), C1 (curve tone 4), B1 (rising tone in sonorant-final syllables – 5a), D1 (rising tone in obstruent-final syllables – 5b), B2 (drop tone in sonorant-final syllables – 6a) and D2 (drop tone in obstruent-final syllables – 6b).

Tone 3 (C1), curve tone ("hỏi"), is expressed by an accent made of the top part of a question (ʔ), e.g. "bả" [ɓa-3] *(residue)*, "chuyển" [tɕwien-3] *(to move)*. It starts on a lowest F0 value among 8 tones and "varies between a High-Falling-Rising realization and a Falling realization with final laryngealization" (Michaud, 2004). "In final syllables, and especially in citation forms, this is followed by a sweeping rise at the end, and for this reason it is often called the "dipping" tone. However, non-final syllables seem only to have a brief level portion at the end, and this is exceedingly elusive in rapid speech" (Thompson, 1987). Although tone 3 is usually described as a low falling and then rising tone, not all Vietnamese speakers have the rising part. The curve tone starts with modal voice phonation, which moves increasingly toward tense voice with accompanying harsh voice (although the harsh voice seems to vary according to speaker).

Tone 4 (C2), broken tone ("ngã"), is written as a tilde (˜), e.g. "bã" [ɓa-4] *(residue)*, "quẫn" [kwɤ̆n-4] *(muddle)*. It starts higher than the falling tone (2), even the level tone (1) and rising. The broken tone has "medial glottal constriction and ends on a high F0 value" (Michaud, 2004). It is accompanied by the rasping voice quality occasioned by tense glottal stricture. Curve and broken tones are "both tense but their tension is not alike and is not distributed across the syllable in the same way" (Nguyen and Edmondson, 1998).

As for the rising tones ("sắc"), they are symbolized by the acute accent (´). Tone 5a (B1), the rising tone in sonorant-final syllables, e.g. "bá" [ɓa-5a] *(residue)*, "choáng" [tɕwaŋ-5a] *(swanky)*, starts higher than the falling tone (2) but lower than the level tone (1). It trails

Table 2.6 – Vietnamese tones

Tone		Name		Register	F0 contour	Duration	Phonation
1	A1	Ngang	Level	High-Mid	Level	Long	Modal
2	A2	Huyền	Falling	Low	Slightly Falling	Long	Lax
3	C1	Hỏi	Curve	Low	Falling	Long	Tense
4	C2	Ngã	Broken	High	Falling-Rising	Long	Glottal
5a	B1	Sắc	Rising	High	Rising	Long	Modal
5b	D1	Sắc	Rising	High	Sharply Rising	Short	Tense
6a	B2	Nặng	Drop	Low	Dropping	Short	Glottal
6b	D2	Nặng	Drop	Low	Sharply Dropping	Short	Tense

upward and rising at the middle of syllables. Phonologically, tone 5a is produced with modal voice (Michaud, 2004). Tone 5b (D1) is the rising tone in obstruent-final syllables, e.g. "bát" [ɓat-5b] *(bowl)*, "bắp" [băp-5b] *(muscle)*, "bách" [bĕk-5b] *(cypress)*, "bác" [bak-5b] *(elder uncle)*. This tone starts on a highest F0 values among 8 tones and sharply rises. Tone 5b is tense and much shorter than other tones (Thompson, 1987)

Tone 6a (B2) and tone 6b (D2), drop tone ("nặng"), are represented by a subscript dot (.). The drop tone is much shorter than other tones with a tendency to go lower. Tone 6a (B2) is the drop tone in sonorant-final syllables, e.g. "bạ" [ɓa-6a] *(strengthen)*, "thường" [tʰuoŋ-6a] *(frequent)*. It starts also high, slightly rising at the beginning, then drops very sharply and is almost immediately cut off by a strong glottal "constriction that is distinct from creaky voice" (Michaud, 2004). Syllables bearing tone 6a have the same rasping voice quality as the broken tone 4. Tone 6b (D2) is the rising tone in syllables having final stops /p t k/, e.g. "bạt" [ɓat-6b] *(canvas)*, "bẹp" [bep-5b] *(crushed)*, "bịch" [bik-5b] *(basket)*, "bạc" [bak-6b] *(silver)*. This tone drops a little more sharply than tone 2, but it is never accompanied by the breathy quality of that tone. It is also tense (Thompson, 1987).

From an experimental point of view, the study of Michaud (2004) confirmed that "precise and reliable information on phonation type (voice quality) can be obtained from electroglottography" as well as that "voice quality is a robust correlate of tone in Vietnamese, showing less variability than F0 across reading conditions". His experiment warranted the conclusion that tones 5b and 6b (i.e. the tones of syllables ending in /p t k/) are "not glottalized, either in final or non-final position".

2.4.3 Tonal coarticulation

The above discussions on Vietnamese lexical tones were mostly for static characteristics. They are more complicated in dynamic states, in which syllables are produced in continuous speech, with phonetic coarticulation effects. Tran and Castelli (2010) did analysis on the influence of coarticulation effect on the variations of tones in continuous speech, and the F0 contour generation was proposed based on this influence of both adjacent tones. Brunelle (2003, 2009) reported that although tonal height coarticulation is bidirectional, progressive tonal coarticulation is much stronger than anticipatory coarticulation in Northern Vietnamese.

2.5 Grapheme-to-phoneme rules

Based on the literature review on Vietnamese phonetics and phonology and some further studies, we presented here some main G2P rules needed for building a pronunciation dic-

tionary for Vietnamese (Section 2.7), and building the G2P conversion module of our TTS system (Chapter 5).

2.5.1 X-SAMPA representation

Since IPA symbols are not appropriate representations in computer-based processing, SAMPA (Speech Assessment Methods Phonetic Alphabet) is adopted to work around the inability of text encodings to represent IPA symbols in TTS. In this work, the X-SAMPA [3] inventory (Gibbon et al., 1997), an extension of SAMPA for individual languages, was adopted for coding phonemes in our Vietnamese TTS system. X-SAMPA can cover the entire range of characters in the IPA. Table 2.7 and Table 2.8 illustrate Vietnamese phonemes in both IPA and X-SAMPA for ease of comparison. These mappings were developed on the basic idea of the work Tran (2007b, p.), with a number of adaptions and extensions.

2.5.2 Rules for consonants

Vietnamese consonants have a set of well-defined grapheme-to-phoneme rules for both initial and final positions. Table 2.7 shows the initial and final consonants with graphemes (orthography) and their respective phonemes. Most of the graphemes have direct rules to convert to corresponding phonemes. They are "b, đ, x, s, g, gh, kh, l, v, th, d, gi, r, ph, tr, h, q, k" for initial, "t, p, n, m, nh" for both initial and final positions.

Table 2.7 – Hanoi Vietnamese initial/final consonants: Grapheme (orthography) to phoneme

No.	Graph -eme	Position	Phoneme		No.	Graph -eme	Position	Phoneme	
			IPA	X-SAMPA				IPA	X-SAMPA
1	b	initial only	ɓ	b	14	t	initial, final	t	t
2	đ	initial only	ɗ	d	15	p	initial, final	p	p
3	x , s	initial only	s	s	16	n	initial, final	n	n
4	g, gh	initial only	ɣ	G	17	ch	final after i,ê,a	k̟	k_+
5	kh	initial only	x	x	18	c	final after u,o,ô	k͡p	kp
6	l	initial only	l	l	19	ch,c	final except 17, 18	k	k
7	v	initial only	v	v	20	m	initial, final	m	m
8	th	initial only	tʰ	t_h	21	nh	final after i,ê,a	ŋ̟	N_+
9	d,gi,r	initial only	z	z	22	nh	initial only	ɲ	J
10	ph	initial only	f	f	23	ng,ngh	initial	ŋ	N
11	tr, ch	initial only	tɕ	ts\	24	ng	final after u,o,ô	ŋ͡m	Nm
12	h	initial only	h	h	25	ng	final except 24	ŋ	N
13	c,k,q	initial only	k	k					

The remaining graphemes have well-defined rules with several exceptional cases as below:

- For the grapheme "gi": In initial positions, if it is followed by consonant, "ê" or nothing, "gi" is converted to /zi/; otherwise /z/
- For the grapheme "ng, ngh":
  - In initial positions, "ng, ngh" is converted to [ŋ]
  - In final positions, if the nucleus is a back rounded vowel /u o ɔ/, "ng" is converted to [ŋ͡m]; otherwise [ŋ]

3. Extended Speech Assessment Methods Phonetic Alphabet

- For the grapheme “nh”:
  - In initial positions, “nh” is converted to [ɲ]
  - In final positions, “nh” is converted to [ŋ͡] (“nh” is in final positions if and only if the nucleus is “i”, “ê” or “a”)
- For the grapheme “ch”:
  - In initial positions, “ch” is converted to /tɕ/, “c” is converted to /k/;
  - In final positions, “ch” is converted to [k͡] (“ch” is in final positions if and only if the nucleus is “i”, “ê” or “a”);
- For the grapheme “c”:
  - In initial positions, “c” is converted to /k/;
  - In final positions, if the nucleus is a back rounded vowel /u o ɔ/, “c” is converted to /k͡p/; otherwise /k/.

2.5.3 Rules for vowels/diphthongs

Most vowels and diphthongs also have direct G2P rules, illustrated in Table 2.8. The graphemes “e”, “ê”, “i, y”, “oo”, “ô”, “ơ”, “u”, “ư”, “ă”, “â” can be respectively converted to the vowels [ɛ], [e], [i], [ɔ], [o], [ɤ], [u], [ɯ], [ă] and [ɤ̆]. The diphthong [iə] can be one of the following orthographies: “ia”, “iê”, “yê”, “ya”. The graphemes “ua” or “uô” can be converted to [uə], while “ưa”, “ươ” are the orthographies of the diphthong [ɯə].

Table 2.8 – Hanoi Vietnamese vowels/diphthongs: Grapheme (Orthography) to phoneme

Vowel type	**Grapheme (Orthography)**	**Phoneme**		**Vowel type**	**Grapheme (Orthography)**	**Phoneme**	
		IPA	X-SAMPA			IPA	X-SAMPA
Long vowel	a	a	a	Short vowel	ă, a (au, ay)	ă	a_X
	e	ɛ	E		â	ɤ̆	7_X
	ê	e	e		a (anh, ach)	ɛ̆	E_X
	i, y	i	i		o (ong, oc)	ɔ̆	O_X
	o, oo	ɔ	O	Diphthong	ia, iê, yê, ya	iə	i@
	ô, ôô	o	o		ua, uô	uə	u@
	ơ	ɤ	7		ưa,ươ	ɯə	M@
	u	u	u				
	ư	ɯ	M				

There are only two exceptional vowels, that is “o” and “a”, having more complicated G2P rules as follows:

- **For the grapheme “o”:** if it is followed by “ng” or “c”, the phoneme is [ɔ̆]; otherwise [ɔ]
- **For the grapheme “a”:** if it is followed by “nh” or “ch”, the phoneme is [ɛ̆]; followed by “u” or “y”, the phoneme is [ă]; otherwise [a].

2.6 Tonophone set

In the HMM-based speech synthesis, many contextual features (e.g., phone identity, locational features) are used to build context dependent HMMs. However, due to the exponential increase of contextual feature combinations, model parameters cannot be estimated accurately with limited training data. Furthermore, it is impossible to prepare speech database that includes all combinations of contextual features and there is great variation in the frequency of appearance of each context dependent unit. To alleviate these problems, a decision-tree based context clustering is the most common technique to cluster HMM states and share model parameters among states in each cluster. Each node (except for leaf nodes) in a decision tree has a context related question. Acoustic attributes of phonemes or contextual features are used to build questions, such as *"Is the current part of speech a noun?"*, *"Is the previous phoneme voiced?"*.

Nethertheless, due to the automation in HMM clustering, the semantics of the contextual features (e.g. the importance or the weight of these features) may not be well considered. Hence, some crucial features of Vietnamese, such as lexical tones, may do not have proper priorities in building decision tree, which may lessen the impact of these features on improvement of the synthetic speech quality. To our best of knowledge, in other work or for other tonal languages, lexical tones may be explicitly modeled in a TTS system (Shih and Kochanski, 2000)(Do and Takara, 2003)(Tran and Castelli, 2010). In the Thai language, tone correctness of the synthetic speech may be improved by investigating the structures of the decision tree with tone information in the tree-based context clustering process of the HMM-based training (Chomphan and Kobayashi, 2008)(Chomphan and Chompunth, 2012)(Moungsri et al., 2014).

With the complexity of the eight lexical tones in the Vietnamese tone system (cf. Section 2.4 in Chapter 2), tone modeling for continuous speech has been a challenging problem. In this work, due to the importance of the Vietnamese lexical tones, we proposed a new speech unit – a **"tonophone"**, which takes into account the lexical tone. We assumed that this new speech unit could model tonal contexts of allophones at high level synthesis. This unit was also essential for corpus design as well as automatic labeling since these tasks required a basic speech unit in their processes.

2.6.1 Tonophone

In this work, to build a phone set for the TTS system, allophones – phonetic realization of phonemes – were used. For example, in the final position, the velar stops /ŋ k/, following back rounded vowels /u o ɔ/ are produced as doubly articulated labial-velars [k͡p ŋ͡m], following /i e ɛ̆/ are actually pre-velar [ŋ̟] and [k̟]. As a result, the six phones for the two phonemes /ŋ k/ are [ŋ k k͡p ŋ̟ k̟].

As aforementioned, lexical tones in Vietnamese syllable structure play a typical and distinguishable role to express perfectly a fully-constituted entity from intonational languages e.g. Indo-European languages. The experimental result of Tran et al. (2005) affirmed that the initial consonant does not take part in the construction the tone of the syllable; hence the impact of the Vietnamese tone is only on the rhyme of the syllable. In conclusion, the Vietnamese tone is non-linear or suprasegmental, i.e. covering and adhering to the rhyme of the syllable, while other parts of the syllable are "linear" or segmental, i.e. continuously sequenced distinct segments. The lexical tone appears simultaneously with segmental phonemes of the rhyme to construct a complete structure of the syllable.

Due to the crucial role of Vietnamese lexical tones not only in the bearing syllables, but also in phonemes in their rhymes, a "tonophone", a new speech unit, was proposed as an allophone regarding the lexical tone, and hence adhered to the lexical tone when possible. In other words, to construct tonophones, the lexical tone was adhered to all allophones in the rhyme, while the initial consonant maintained its form without any information of the tone. "Tonophones" were used for emphasizing the role of lexical tones, and reflected their corresponding allophones in tonal contexts. We believed that this new speech unit might give us more precise analysis/design and better synthetic speech.

For instance, a syllable "ngoèo" [ŋwɛw-2] *(in "ngoằn ngoèo" – zigzagging)* carrying the broken tone 2, actually composes [ŋ] as the initial, [^{w2}] as the medial, [ɛ2] as the nucleus and [w2] as the final. Its transcription was [N^{w2}E2w2-2] when representing in tonophones.

2.6.2 Tonophone set

Table 2.9 shows how Vietnamese allophones combines with possible lexical tones to build the tonophone set, based on our literature review in the previous sections. As aforementioned, since the initial consonant does not carry the information of the lexical tone, 19 initial consonants did not adhere to any tone when forming the corresponding tonophones. Whereas, the medial [^{w}] and 16 nucleus (including 3 dipthongs [iə uə ɯɤ]) were combined with 8 tones as these elements appear in both sonorant- and obstruent-ending syllables.

Table 2.9 – Vietnamese tonophone set

Syllable element	Allophones	Lexical tones	Tonophone #
Initial consonant	p ɓ t t^{h} ɗ k m n ɲ ŋ tɕ f v s z x ɣ h l	*(Not adhering)*	19 x 1
Medial	^{w}	1-4, 5a, 5b, 6a, 6b	1 x 8
Nucleus	i a u e ɯ ɤ ɔ ɛ ɤ̆ ɔ̆ ă ɛ̆ ă iə uə ɯə	1-4, 5a, 5b, 6a, 6b	16 x 8
Final consonant	p t k k͡p k̚	5b, 6b	5 x 2
	m n ŋ ŋ͡m ŋ̚ w j	1-4, 5a, 6a	7 x 6
Total			**207**

The obstruent-final syllables may carry either of two tones: the rising tone 5b or the drop tone 6b. Hence, only these two tones (5b, 6b) were embedded to 5 allophones that are unreleased final stops [p t k k͡p k̚]. Whereas, other 7 allophones at the final positions (including 2 semi-vowels [w j]) were combined with 6 tones 1-4, 5a, 6a.

As a result, there are 48 Vietnamese allophones (without considering the lexical tones). A total of 207 tonophones were constructed for the tonophone set for our work.

2.6.3 Acoustic-phonetic tonophone set

An acoustic-phonetic unit set of the target language is an important input for a TTS system, especially for the HMM-based approach. It is intended to represent every speech segment that is clearly bounded from an acoustic point of view, as well as every speech segment that is phonetically significant, even if it is not clearly bounded. The two main usages of this set are in (i) HMM clustering using phonetic decision trees, and (ii) automatic labeling, i.e. automatic segmenting and force-aligning the speech corpus with the orthographic transcriptions.

In the HMM-based speech synthesis, as aforesaid, context clustering is an important process in the training phase to treat the problem of limitation of training data. Acoustic attributes of phones are crucial information to build questions for nodes and to construct decision trees. The construction of decision trees makes a great contribution to improve the quality of synthetic speech.

With the increase in size of speech databases, manual phonemic segmentation and labeling of every utterance became unfeasible. Thus, automatic labeling was one important task to build an annotated corpus for TTS systems. The acoustic-phonetic unit set is used to model and to identify clear acoustic events, which an expert phonetician would mark as boundaries in a manual segmentation session. Therefore, it is essential to obtain a good labeling.

Table 2.10 – Hanoi Vietnamese acoustic-phonetic tonophones – consonants

No.	Consonant	Place	Manner	Voicing
1	b	Plosive	Bi-labial	Voiced
2	d	Plosive	Alveolar	Voiced
3	t^h	Plosive	Dental	Voiceless
4	p, p5b, p6b	Plosive	Bi-labial	Voiceless
5	t	Plosive	Dental	Voiceless
6	t5b, t6b	Plosive	Alveolar	Voiceless
7	k, k5b, k6b	Plosive	Velar	Voiceless
8	k̟{5b, 6b}	Plosive	Pre-velar	Voiceless
9	k͡p{5b, 6b}	Plosive	Labial-velar	Voiceless
10	f	Fricative	Labio-Dental	Voiceless
11	v	Fricative	Labio-Dental	Voiced
12	h	Fricative	Glottal	Voiceless
13	x	Fricative	Velar	Voiceless
14	ɣ	Fricative	Velar	Voiced
15	s	Fricative	Alveolar	Voiceless
16	z	Fricative	Alveolar	Voiced
17	tɕ	Affricative	Palatal	Voiceless
18	m, m{1-4,5a,6a}	Nasal	Bi-labial	Voiced
19	n	Nasal	Dental	Voiced
19	n{1-4,5a,6a}	Nasal	Aveolar	Voiced
20	ŋ, ŋ{1-4,5a,6a}	Nasal	Velar	Voiced
21	ŋ̟{1-4,5a,6a}	Nasal	Pre-velar	Voiced
22	ŋ̟͡m{1-4,5a,6a}	Nasal	Labial-velar	Voiced
23	ɲ	Nasal	Palatal	Voiced
24	l	Lateral-approximant	Dental	Voiced
25	w{1-4,5a,6a}	Approximant	Labial-dental	Voiced
26	j{1-4,5a,6a}	Approximant	Dental	Voiced

Based on the phonetics and phonological system of Vietnamese, an acoustic-phonetic unit set for Vietnamese was built with main phonetic attributes for both consonants and vowels. For consonants, the attributes were: (i) place or articulation, i.e. labial, labio-dental, alveolar, retroflex, palatal, labial-velar, dental, and velar, (ii) manner of articulation, i.e. nasal, stop/plosive, fricative, affricative, approximant, and lateral-approximant, and (iii) voicing, i.e. voiced and voiceless. The phonetic attributes for vowels included: (i) position of tongue, i.e. front, central, and back, (ii) height, i.e. close (high vowels), close-mid, open-mid, and open (low vowels), (iii) length, i.e. short, long, and diphthong, and (iv) roundedness, i.e. rounded

Table 2.11 – Hanoi Vietnamese acoustic-phonetic tonophones – vowels

No.	Vowel	Position	Height	Length	Roundedness
1	i{1-4,5a,5b,6a,6b}	Front	Close	Long	Unrounded
2	ɯ{1-4,5a,5b,6a,6b}	Back	Close	Long	Unrounded
3	u{1-4,5a,5b,6a,6b}	Back	Close	Long	Rounded
4	e{1-4,5a,5b,6a,6b}	Central	Close-mid	Long	Unrounded
5	ɤ{1-4,5a,5b,6a,6b}	Back	Close-mid	Long	Unrounded
6	ɤ̆{1-4,5a,5b,6a,6b}	Back	Close-mid	Short	Unrounded
7	o{1-4,5a,5b,6a,6b}	Back	Close-mid	Long	Rounded
8	ɛ{1-4,5a,5b,6a,6b}	Front	Open-mid	Long	Unrounded
9	ɛ̆{1-4,5a,5b,6a,6b}	Front	Open-mid	Short	Unrounded
10	a{1-4,5a,5b,6a,6b}	Front	Open	Long	Unrounded
11	ă{1-4,5a,5b,6a,6b}	Front	Open	Short	Unrounded
12	ɔ{1-4,5a,5b,6a,6b}	Back	Open-mid	Long	Unrounded
13	ɔ̆{1-4,5a,5b,6a,6b}	Back	Open-mid	Long	Unrounded
14	iə,ɯə{1-4,5a,5b,6a,6b}	Central	Close	Diphthong	Unrounded
16	uə{1-4,5a,5b,6a,6b}	Central	Close	Diphthong	Rounded

and unrounded.

We assumed that the phonetic features of phones and tonophones were similar. Based on the literature review in Section 2.3, a complete acoustic-phonetic tonophone set of Hanoi Vietnamese was built and is illustrated in Table 2.10 (consonants) and Table 2.11 (vowels).

2.7 PRO-SYLDIC, a pronounceable syllable dictionary

There was a need to build a syllable e-dictionary whose entries are syllables with their transcriptions. This dictionary was used in natural language processing (high-level speech synthesis) in our TTS system. The main purpose of this dictionary was used for transcribing Vietnamese text. It could be also used for filtering pronounceable syllables in order to extract non-standard words (i.e. tokens that cannot be directly transcribed to phonemes, e.g. numbers, dates, abbreviations).

Therefore, pairs of syllable orthography and transcription were automatically generated mainly based on (i) the G2P rules (Section 2.5), (ii) syllable-forming orthographic rules, and (iii) the list of rhymes in Table 2.12. Tonophones were used as a speech unit in the transcriptions, such as “nhuyễn” /ɲw4ie4n4-4/ *(fine)*. However, for the simplicity, transcriptions in this section were represented in allophones without regard to lexical tones, such as “nhuyễn” /ɲwien-4/ *(fine)*.

2.7.1 Syllable-orthographic rules

Some following syllable-orthographic rules were found in the Vietnamese language. These rules were essential to build the list of rhymes and the PRO-SYLDIC.

- **For initial consonants:**
  - **I1**: “ngh”, “ng” (/ŋ/); “gh”, “g” (/ɣ/); or “k”, “c” (/k/): if the nucleus is /i e ɛ/, the initial consonant is “ngh”, “gh”, or “k” respectively; otherwise, it is “ng”, “g”, “c” respectively;

  - **I2**: The labial onsets are never accompanied by a secondary labial articulation [ʷ], for example "hoa" [hwa] *(flower)* exists but "boa" [ɓwa] does not in the language.

- **For medial (pre-tonal) sounds /ʷ/:**
  - **M1**: The orthography "u" either follows the grapheme "q" or precedes narrow/quite-narrow nucleus, i.e. /i e ɤ ɤ̆ iə/, i.e. "i", "y", "ê", "ơ", "â", "yê", "ya", e.g. "(q)uang", "uyêt", "uya", "(q)uyt", "uơ", "uân";
  - **M2**: The orthography "o" always precedes open or open-mid vowels, i.e. /ɛ a ă/, e.g. "oe", "oan", "oăt".

- **For final consonants:**
  - **F1**: The semi-vowel /j/ never follows the front nucleus /i iə e ɛ/, while /w/ never follows rounded nucleus /u uə o ɔ/
  - **F2**: The orthography of the semi-vowel /w/ is "o" if the nucleus is /a/ or /ɛ/, "u" for other cases, e.g. "ao", "eo", "âu", "iu";
  - **F3**: The orthography of the semi-vowel /j/ is "y" if the nucleus is /ă/ or /ɤ̆/; "i" for other cases, e.g. "ay", "ây", "ai", "ui";
  - **F4**: The orthography of the stop /k/ is "ch" if the nucleus is /i/, /e/ or /ĕ/, "c" for other cases, e.g. "ích", "ếch", "ách", "ác", "ắc", "iếc".

2.7.2 Pronounceable rhymes

As presented in Section 2.2, the eight structure-based types of syllables are: (i) nucleus alone, (ii) initial+nucleus, (iii) medial+nucleus, (iv) nucleus+ending, (v) initial+medial+nucleus, (vi) initial+nucleus+ending, (vii) medial+nucleus+ending, and (viii) initial+medial+nucleus+ending. Rhymes was concluded to compose of medial, nucleus and ending, hence support four types: (i) nucleus alone, (ii) medial+nucleus, (iii) nucleus+ending, and (iii) medial+nucleus+ending. A syllable may optionally contain an obstruent, nasal, or approximant coda. The structure of rhymes is (ʷ)V(C), where ʷ is the glide [ʷ], V is a vowel or a diphthong, C is a final consonant, which can be one of /p t k ŋ m n/ or a semi-vowel /j w/, and T is a tone (1-4, 5a, 5b, 6a, 6b).

Table 2.12 presents a phonetic analysis of pronounceable rhymes in Hanoi Vietnamese. This table was created from the idea of Michaud et al. (2015) for Phong Nha dialect of Vietnamese. It was developed using our review on the Hanoi Vietnamese phonetics and phonology. For ease of representation, the phonetic realizations of /k/ or /ŋ/, i.e. [k͡p̚ k̟̚] or [ŋ͡m̟ ŋ̟], are also located in the same lines as /k/ or /ŋ/ respectively. Due to the limited space, a haft of nucleus (i.e. /i e ɛ ĕ iə ɯ ɤ ɤ̆/) can be observed in the first haft rows of the table (10 first rows including headers), while others (i.e. /u o ɔ ŏ uə ɯə a ă/) can be found in the second part of the table (10 last rows including headers). The first column shows the structure of rhymes, in which "V" denotes a vowel or a diphthong. Each row presents possible rhymes without (left) and with (right) the glide medial [ʷ] for each structure differentiating from the final consonant (the absence of final consonant or one of /k ŋ t n p m j w/). For example, the rhyme "oen" [ʷɛn] is located in the right (because of the medial [ʷ]) of the "ɛ" column (hence the first half of the table), and the "Vn ʷVn" row.

As presented as an orthographic rule for the medial (the rule M2), the orthography "o" always precedes open or open-mid vowels, i.e. /ɛ a ă/, e.g. "oe", "oac", "oăn". However,

if the initial consoant is /k/, its orthography must be "q". In this case, the orthography "o" for the medial [ʷ] must be replaced by "u". For instance, from the rhyme "oac", an example of a syllable with the initial consonant /k/ is "quác" [kʷak-5b] *(quack)*, while the one with the initial consonant "t" is "toác" [tʷak-5b] *(cleave)*. The following rhymes exist in Vietnamese: "(q)ue, (q)uet, (q)uen, (q)ueo, (q)ua, (q)uac, (q)uang, (q)uanh, (q)uach, (q)uat, (q)uan, (q)uay, (q)uăc, (q)uăng, (q)uăt, (q)uăn, (q)uăp, (q)uăm, (q)uau, (q)uao, (q)uai", whereas "(q)uec, (q)ueng, (q)uep, (q)uem, (q)uap, (q)uam" are pronounceable but do not exist in the language.

The main difference of this table from previous studies was that there are some nonexistent yet pronounceable rhymes (with a star * in the right). For example, the rhyme "oep" does not appear in any meaningful Vietnamese syllables/words, however, based on the Vietnamese G2P rules in Section 2.5, this can be transcribed to [ʷep]. The reason to maintain these rhymes is that the input of a Vietnamese TTS system may include numerous loanwords that includes nonexistent but pronounceable syllables, as well as newly appeared words from teenagers or Internet users.

For instance, the vietnamese-style pronounce for the word "website" may be "goép sai" [ɣʷep-5b saj-1] or sometimes "oép sai" [ʷep-5b saj-1] depending on the speakers. In fact, "goép" or "oép" does not exist in Vietnamese. As aforesaid, the list of rhymes was used for generating the syllable dictionary for transcribing Vietnamese text input, which may be from a number of sources (e.g. Internet, stories). Those rhymes provided a great mean to transcribe all pronounceable syllables for a TTS system. A dash (–) indicates that the combination at issue is not pronounceable in the language.

Table 2.12 – Hanoi Vietnamese pronounceable rhymes, *: not exist but pronounceable. The medial orthography "o" (e.g. "oanh" [ʷɛŋ͡]) is changed to "u" if the initial is /k/ (its orthography must be "q"), e.g. "l**o**anh q**u**anh" [lʷɛŋ͡ qʷɛŋ͡] *(to go around)*; some rhymes do not exist yet are pronounceable: (q)uec, (q)ueng, (q)uep, (q)uem, (q)uap, (q)uam

Structure	i	e	ɛ	ɛ̆	iə	ɯ	ɤ	ɤ̆
V ʷV	i uy	ê uê	e oe	– –	ia uya	ư –	ơ uơ	– –
Vk ʷVk	ich uych	êch uêch	ec oec*	ach oach	iec –	ưc –	– –	âc –
Vŋ ʷVŋ	inh uynh	ênh uênh	eng oeng*	anh oanh	iêng –	ưng –	ơng* –	âng uâng
Vt ʷVt	it uyt	êt uêt	et oet	– –	iêt –	ưt –	ơt –	ât uât
Vn ʷVn	in uyn	ên uên	en oen	– –	iên uyên	ưn –	ơn –	ân uân
Vp ʷVp	ip uyp	êp uêp*	ep oep*	– –	iêp uyêp*	ưp* –	ơp –	âp uâp*
Vm ʷVm	im uym*	êm uêm*	em oem*	– –	iêm uyêm*	ưm –	ơm –	âm uâm*
Vj ʷVj	– –	– –	– –	– –	– –	ưi –	ơi –	ây uây
Vw ʷVw	iu uyu	êu –	eo oeo	– –	iêu –	ưu –	– –	âu uâu*

Structure	u	o	ɔ	ɔ̆	uə	uɯə	a	ă
V ʷV	u –	ô –	o –	– –	ua –	ưa –	a oa	– –
V ʷV	uc –	ôc –	ooc –	oc –	uôc –	ươc –	ac oac	ăc oăc
Vŋ ʷVŋ	ung –	ông –	oong –	ong –	uông –	ương –	ang oang	ăng oăng
Vt ʷVt	ut –	ôt –	ot –	– –	uôt –	ươt –	at oat	ăt oăt
Vn ʷVn	un –	ôn –	on –	– –	uôn –	ươn –	an oan	ăn oăn
Vp ʷVp	up –	ôp –	op –	– –	uôp –	ươp –	ap oap	ăp oăp
Vm ʷVm	um –	ôm –	om –	– –	uôm –	ươm –	am oam	ăm oăm
Vj ʷVj	ui –	ôi –	oi –	– –	uôi –	ươi –	ai oai	ay oay
Vw ʷVw	– –	– –	– –	– –	– –	– –	ao oao	au oau*

2.7.3 PRO-SYLDIC

The total number of rhymes in Table 2.12 is 170, in which 62 rhymes ending in /p t k/ (called sonorant-final rhymes). As aforementioned, sonorant-final syllables can only carry the rising

and drop tones (5b, 6b), the 62 sonorant-final rhymes can also bear these two tones, making a total of 124 sonorant-final rhymes with tones. The 108 obstruent-final rhymes can carry six tones (1-4, 5a, 6b), making a total of 648 obstruent-final rhymes with tones.

The purpose of the PRO-SYLDIC (PROnounceable SYLlable DICtionary) was to build the transcriptions for all pronounceable syllables in Vietnamese, hence all 19 initial consonants were combined with a total of 772 rhymes with tones. There did exist many nonexistent syllables in some combinations, however they were useful to transcribe loanwords or newly appeared syllables. For example, a loanword "boa" [ɓwa] in 'tiền boa' from French 'pourboire' *(tip)* appeared sometimes in the real input text although it does not exist in the language due to the violation of the rule I3 "*The labial onsets are never accompanied by a secondary labial articulation [ʷ]*".

As a result, there were totally 21,648 syllables with tones (orthography) in the PRO-SYLDIC, including rhymes without initial consonants. Some of the complexities of the orthography in the language were also covered. For instance, consider combinations of the initial consonant /k/ with some rhymes. Due to the rule I2 (the orthography of /k/ is "c" if the nucleus is not "i", "e", "ê", or "y"), the rhyme "ua" [uə] (the second haft of the table: the "uə" column and the "V ʷV" row) is combined with "c" to become the syllable **"cua"** [kuə**-1**] *(crab)*. Meanwhile, the syllable with the initial /k/ of the rhyme "oa" [ʷa] (the second haft of the table: the "a" column and the "V ʷV" row) is **"qua" [k^{w}a-1]** *(pass away)*.

2.8 Conclusion

This chapter presents our literature review was done on (i) the syllable structure, (ii) the phonological system, and (iii) the lexical tones for Hanoi Vietnamese, a sort of standard Vietnamese. In the hierarchical structure of Vietnamese syllable, lexical tones, a non-linear or suprasegmental part, appear simultaneously with segmental elements of rhyme, i.e. medial, nucleus and ending. There are 19 initial consonants and 12 phones in the final position. Hanoi Vietnamese distinguishes one medial rounding glide, nine long vowels, four short vowels, and three falling diphthongs. The Vietnamese tone system, which belongs to the pitch-plus-voice quality type, has (i) a six-tone paradigm for sonorant-final syllables: level tone 1 (A1), falling tone 2 (A2), curve tone 3 (C1), broken tone 4 (C2), rising tone 5a (B1), and drop tone 6a (B2); and (ii) a two-tone paradigm for obstruent-final syllables: rising tone 5b (D1), drop tone 6b (D2). The broken tone 4 has medial glottal constriction while the rising tone 6a drops very sharply and are almost immediately cut off by a strong glottal constriction at the end. The two tones 5b and 6b are not glottalized, either in final or non-final position.

Based on the literature study, several tasks were performed for building our TTS system as well as for designing a new corpus.

First, grapheme-to-phoneme rules were developed for transcribing Vietnamese consonants and vowels/diphthongs. Many graphemes can be directly converted to phones without any ambiguity, such as "b-" to [ɓ], "ch-, tr-" to [tɕ], "-m" to [m], "ê" to [e]. Well-defined rules were found for more complicated cases/variants. For instance, for the grapheme "a", if it is followed by "nh" or "ch", the phoneme is [ɛ̆]; if it is followed by "u" or "y", the phoneme is [ă]; otherwise, the phoneme is [a]. The full G2P rules were used for both transcribing the raw text for corpus design and building the G2P conversion module of our TTS system.

Second, due to the great importance of lexical tones, a "tonophone" – an allophone in tonal context, was proposed as a new speech unit for our work. In this research, to build the tonophone set of the system, the lexical tone was taken into account and adhered to all allophones in the rhyme, and the initial consonant maintained its form without any information

of the tone. As a result, a tonophone set with 207 tonophones was constructed from 48 Vietnamese allophones. This unit set includes: (i) 19 initial consonants without tone information, (ii) medial and 16 nucleus adhering to eight tones, (iii) unreleased final stops adhering to two tones 5b, 6b, and (iv) other final consonants adhering to six tones 1-4, 5a, 6a. An acoustic-phonetic tonophone set of Vietnamese was also built for (i) HMM clustering using phonetic decision trees, and (ii) automatic labeling, i.e. automatic segmenting and forced aligning the speech corpus with the orthographic transcriptions. Based on the literature review, main phonetic attributes were specified for both consonants and vowels on this acoustic-phonetic unit set, such as place or articulation or manner of articulation for consonants, position of tongue or height for vowels.

And finally, PRO-SYLDIC, a Vietnamese syllable transcription e-dictionary was constructed for filtering pronounceable syllables in text normalization as well as transcribing texts. Pairs of syllable orthography and transcription in the dictionary were automatically generated mainly based on (i) the G2P rules, (ii) the syllable-orthographic rules, and (iii) the list of rhymes. A table of 170 Vietnamese rhymes with not only existent but also pronounceable ones in the language was designed. The reason to maintain all pronounceable rhymes is that the input of a Vietnamese TTS system may include numerous loanwords that includes nonexistent but pronounceable syllables, as well as newly appeared words from teenagers or Internet users. The PRO-SYLDIC was constructed by combining 19 initial consonants with 772 rhymes bearing tones, making a total of 21,648 pronounceable syllables (orthography).

Chapter 3

Corpus design, recording and pre-processing

Contents

3.1 Introduction

Over the last two decades, with the rise of corpus-based speech synthesis (e.g. unit selection and HMM-based speech synthesis), speech database has greatly contributed to the quality of synthetic voice. This leads to the necessity of providing a proper speech corpus for the TTS system training and testing.

Several works on Vietnamese speech corpus were presented, but mainly for speech recognition (Le et al., 2004, 2005, Vu and Schultz, 2009, 2010, Vu et al., 2005). These works did not focus on designing the text corpus, but on collecting/recording the speech corpus, as well as on selecting speakers or on automatic alignment.

The VNSP corpus (short for "VNSpeechCorpus for synthesis") of the unit selection TTS system of Tran (2007b) was collected from different resources from the Internet (e.g. stories, books, web documents), and manually chosen by experts. It included various types of data: words with six lexical tones, figures and numbers, dialog sentences and short paragraphs. It comprised about 630 sentences in 37 minutes, recorded by a TV broadcaster from Hanoi. The work of Vu et al. (2009) reported that a 3000-sentence training corpus was constructed according to a set of phonetically-rich sentences for spoken Vietnamese. However, to the best of our knowledge, there lacks a thorough work on analysis and design of a text corpus for the Vietnamese TTS, especially for the HMM-based approach.

A number of papers have targeted at text corpus design for speech processing in various languages. Many researchers proposed methods to design a phonetically balanced corpus, such as Uraga and Gamboa (2004) for Mexican Spanish, Oh et al. (2011) for Korean (speech coding), or Abushariah et al. (2012) for Arabic. Some used a phonotactic approach to design a text corpus with a full coverage of phonemes and allophones in every possible context (Uraga and Gamboa, 2004), or even used an enormous phonetically rich and balanced source from Web (Villaseñor-Pineda et al., 2004). Due to the special requirement of speech coding, the work of Oh et al. (2011) proposed a method based on a similarity measure, which calculated how close the phoneme distribution occurring from natural conversation was to that of the designed text corpus. Most of the works adopted the greedy search algorithm to select the best candidate sentences. Some others considered the design of speech database for TTS systems as a set-covering problem (Chevelu et al., 2008, Francois and Boëffard, 2001). It seems to us that the greedy algorithm is robust and reliable enough to design a corpus for a general TTS system, with unlimited vocabulary as our initial motivation.

Other tonal languages, such as Mandarin Chinese, were also investigated on the corpus design for data-driven TTS systems (Chou et al., 2002, Tao et al., 2008, Zhu et al., 2002). The corpus of Tao et al. (2008) was delivered to Blizzard Challenge 2008 as the common corpus for the Mandarin speech synthesis evaluation among all participants. 5,000 phonetic context balanced sentences were finally chosen by an automatic prompt selection with the greedy search algorithm and several criterions from raw text. Syllables were considered as a speech unit in the design with 12 factors for the prompt selection, including the previous, current and next lexical tones.

This chapter describes our proposal on the corpus design for the Vietnamese TTS systems including for the HMM-based approach. The initial motivation was to design a phonologically-rich and -balanced corpus in both phonemic and tonal contexts. However, to manually build such a corpus consumes a great human effort for the selection task. Section 3.2 describes a huge raw text, which was crawled from different sources from the Web and other sources. Section 3.3 presents the processing of the raw text (e.g. text cleaning, text normalization, text transcription) to be ready for the further tasks.

As aforesaid, since Vietnamese is a tonal language, speech units in this work were considered not only in phonemic context but also in tonal context. Section 3.4 provides an analysis of two new speech units: (i) a "tonophone": an allophone adhering with a lexical tone when possible (cf. Section 2.6 in Chapter 2), and (ii) a "di-tonophone": an adjacent pair of "tonophones". Our work targets to design a phonetically-rich and -balanced corpus in terms of tonophones and di-tonophones. To provide a reference for the design process, the phonetic distributions of the raw text are described in this section.

Section 3.5 shows the design process and results of several corpora using the greedy algorithm and all information from above preparations. The recording environment and quality control of resulting corpora were described in Section 3.6. Some treatments (e.g. automatic labeling, correcting breath noises) on speech corpus for TTS systems are finally given in Section 3.7.

3.2 Raw text

3.2.1 Rich and balanced corpus

In a TTS system, a speech corpus plays an important role in generating good acoustic models, hence producing a high-quality synthesizer. In some systems, such as concatenative systems, if there lack some essential acoustic units to synthesize a specific sentence, the quality of the synthetic speech will be degraded. Although HMM-based speech synthesis is more robust if the phonetic balances of the text are not ideal, a poor training corpus may cause a bad or even unintelligible synthetic speech. As a result, the speech corpus for a TTS system, especially with unlimited vocabulary, must be phonetically-**rich** and -**balanced** (Villaseñor-Pineda et al., 2004) (Abushariah et al., 2012).

To address the first criterion, a speech corpus can be considered a **phonetically rich** one if it contains all the phones and has a good coverage of other speech units (e.g. di-phones, triphones) of the language. In other words, it should provide a good coverage of one or several speech units. A full coverage of phones ensures their availability for the synthesis process, which helps a TTS system produce an intelligible synthetic speech. A speech corpus with a good coverage of bigger speech units, e.g. di-phones or triphones, provides more suitable templates in different contexts. This ameliorates the quality of synthetic speech: more intelligible, more natural, etc.

A **phonetically balanced** corpus maintains the phonetic distribution of the language. In other words, if phone a has a higher frequency than phone b in the language, it should appear more often than phone b in the speech corpus. With such a corpus, a more common sentence would be synthesized with better quality than the less common one. As a result, the quality of the TTS system will be improved at least for common cases in spite of a finite amount of corpus.

3.2.2 Raw text from different sources

To manually build a phonetically-rich and -balanced corpus consumes a great human effort for the selection task. In this work, a big raw text was crawled from the Web and other sources. This resource could be considered as a phonetically-rich and balanced source, and might represent for the Vietnamese language in terms of phonetic distribution due to its variety and its large size. A variety of sentence types, sentence modes, sentence lengths, etc. may introduce the corpus a vast range of context, hence improve the quality of TTS systems. Therefore, five major sources were used for building a big raw text: (i) e-newspapers,

(ii) e-stories, (iii) existing sources, (ii) Vietnamese e-dictionary (VCL), and (v) special design.

E-newspapers. The two main sources of e-newspapers came from: (i) the project for the blind "Tâm hồn Việt Nam" (Vietnamese souls), and (ii) the training corpus for a Vietnamese-French statistical machine translation system (Do et al., 2009). One of the most important tasks in the "Vietnamese souls" project was to automatically crawl from different e-newspapers to a unique source [1] for the Vietnamese blind. The final resource for our work from this project was extracted from different topics of some well-known e-newspapers (such as http://dantri.com.vn, http://vnexpress.net, http://vietnamnet.vn). There were a total of 3,795 articles with 132,514 sentences. The work of Do et al. (2009) proposed a document alignment method for mining a comparable Vietnamese-French corpus. The first result contained about 12,100 parallel document pairs and 50,300 parallel sentence pairs. However, we obtained about 142,305 Vietnamese sentences in the final bilingual corpus.

E-stories and existing sources. Seven e-stories collected from web pages provided us about 13,856 sentences in paragraphs, and 20,523 sentences in dialogs. Existing resources included 630 sentences of the VNSP corpus (i.e. VNSpeechCorpus for synthesis - the old one), 10,368 sentences of the VietTreebank, 5,000 sentences from Vietnamese Wikipedia.

The VCL dictionary and special design. The VCL dictionary is a Vietnamese e-dictionary for natural language processing of the VLSP project, hosted by the Vietnam Lexicography Centre (Vietlex). The VCL dictionary comprised about 35,000 Vietnamese words with their definitions and examples in the XML-based structure for ease of use. The examples in VCL were sentences (usually short) containing the target words. Some words either had examples with incomplete-sentences (i.e. words or phrases) or were lack of examples. Special design was done for these cases.

3.3 Text pre-processing

3.3.1 Main tasks

As mentioned in Section 3.2, we design a synthesis corpus by selecting the richest and most balanced sentences from a raw text, which may represent for the language in terms of phonetic distribution. This resource was huge and was collected from various sources, hence there was a need to pre-process to make it to be suitable for the design process. Figure 3.1 illustrated the procedure of raw text pre-processing, including five main tasks: (i) Sentence segmentation, (ii) Tokenization, (iii) Text cleaning, (iv) Text normalization, and (v) Text transcription. First, texts were segmented into sentences for further treatments. These sentences were tokenized into syllables or Non-Standard Words (NSWs – which cannot be directly transcribed to phonemes, e.g. numbers, dates, abbreviations, currency). Each sentence was then examined to be not "clean" or "too long" for removing. The three first tasks of the text pre-processing were performed during the text collection for ease of storing and management. The next task was text normalization, in which NSWs were then processed and expanded to speakable syllables. Finally, the normalized text was transcribed into tonophones to provide a suitable input for next steps. The details of these tasks will be described in the next subsections.

1. http://tamhonvietnam.net

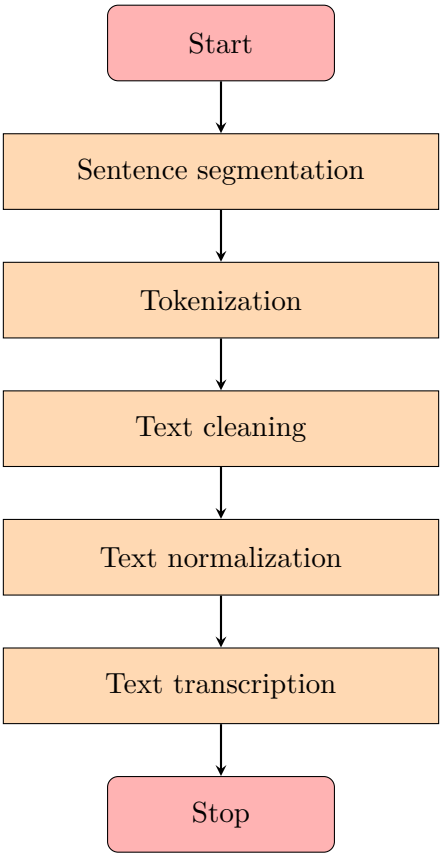


Figure 3.1 – Main tasks in raw text pre-processing.

3.3.2 Sentence segmentation

Text from these resources needed to be segmented into sentences for ease of management. Regular expressions were mainly used for segmenting sentences. Some ambiguous cases were separately treated with particular strategies or heuristics. Some had to be manually corrected. Each sentence was then split into tokens (syllables, abbreviations, etc.) by spaces, punctuations, etc. Each sentence was assigned a unique code, including two parts: (i) four letters indicating the sources: NEWS (e-newspapers), STOR (e-stories), VCLD (VCL Dictionary), VLSP (VNSpeech corpus for synthesis), WIKI (Vietnamese wiki), SPEC (special design); and (ii) six digits indicating its position in each source, starting from 1: 000001, 000002, etc. The sentence number of each source in the raw text presented in Section 3.2 was calculated after this task.

The total sentence number of the raw text was 349,095 sentences from five major sources.

3.3.3 Tokenization into syllables and NSWs

As aforementioned, Vietnamese is an isolating language, in which the boundary of syllable and morpheme is the same, each morpheme is a single syllable. Each syllable usually has an independent meaning in isolation, and polysyllables can be analyzed as combinations of monosyllables (Doan, 1977). Hence, a syllable in Vietnamese is not only a phonetic unit but also a grammatical unit (Doan, 1999b). Besides, Vietnamese text is actually a sequence of syllables, separated by spaces. As a result, each sentence had to be tokenized into syllables

and NSWs. Spaces and punctuations were used as the best delimiters for this task.

3.3.4 Text cleaning

Since the raw text was collected from various sources, mainly from the Web, there existed "unclean" or unsuitable sentences that cannot be used in designing corpus. Sentences having more than 70 syllables were considered "very long" sentences and hence were removed from the raw text. Sentences having unreadable (e.g. control) symbols or wrong encoding were also removed. After text cleaning, the raw text bank of 349,095 sentences was reduced to 323,934 "clean" sentences, which means that about 7% were unsuitable and removed.

Table 3.1 – The final raw data for Vietnamese corpus design

Source	Sentence # in paragraphs	Sentence # in dialogs	Syllable #	Mean length (syllables/sentence)
E-newspapers	255,145	0	9,432,669	37.0
E-stories	13,601	20,402	450,655	13.3
VCL dictionary	15,600	3,900	155,274	8.0
VNSP	433	197	8,930	14.1
VietTreebank	7,723	1,836	216,725	22.7
Wiki	4,146	793	109,373	22.1
Special design	78	117	2,905	14.9
TOTAL	**296,704**	**27,230**	**10,377,903**	**32.0**

Table 3.1 shows some information of various sources in the final raw text. E-newspapers occupied the highest proportion (about 91%) of the raw text. The mean of sentence length (number of syllables) was also the largest, about 37 syllables per sentence, and only contained sentences in paragraphs, not in dialogs. Example sentences (for Vietnamese words) in the VCL dictionary were short in general, averaging 8 syllables per sentence. E-stories included more dialogs than paragraphs (about 1.5 times), hence about 13.3 syllables per sentence on average (dialogs includes short sentences in general). Other resources ranged from 14 syllables to 23 syllables per sentences. They contained more paragraphs than dialogs. The minimum sentence length of the raw text was 1 syllable, and the average sentence length was 32 syllables. There are total more than 10 billions syllables in the raw text.

3.3.5 Text normalization

The Vietnamese real text included Non Standard Words (NSW), which cannot be directly transcribed to phonemes, e.g. numbers, dates, abbreviations, currency. The pronunciation of these NSWs cannot be found by applying "letter-to-sound" rules. Such NSWs include numbers; digit sequences (such as telephone numbers, date, time, codes...); abbreviations (e.g. "ThS" for "Thạc sĩ"); words, acronyms and letter sequences in all capitals (e.g. "GDP"); foreign proper names and place names (such as "New York"); roman numerals; URL's and email addresses. Normalization of such words, called **text normalization**, is the process of generating normalized orthography from text containing NSWs.

The text normalization process adopted the main idea from our previous work (Nguyen et al., 2010), which normalized NSWs to the appropriate form so that it became speakable. However, due to the vast and various sources, only basic processing tasks were done on the raw text.

These NSWs first were identified by filtering tokens using the PRO-SYLDIC dictionary (cf. Section 2.6 in Chapter 2). Those candidates were then classified into corresponding categories using regular expressions.

These NSWs were expanded to full text according to their categories. Numbers, dates, times, currencies, measures, etc were expanded by well-defined rules. For example, a date "13/04/1994" in Vietnamese (the format "dd/mm/yyyy") was expanded to "ngày mười ba tháng tư năm một nghìn chín trăm chín mươi tư" *(day thirteen month fourth year one thousand nine hundreds ninety four)* by the following rules:

- A date starts with the word "ngày" *(day)*;
- The day is expanded to a normal number; if the day is smaller than 11, it is preceded by "mùng";
- The month is expanded to a normal number; except that "4" or "04" is expanded to "tư" *(forth)*;
- The year is expanded to a normal number.

Abbreviations were expanded by looking up an abbreviation dictionary (435 entries), such as "ĐHBKHN" was expanded to "Đại học Bách Khoa Hà Nội" *(Hanoi University of Science and Technology)*, "CLB" was expanded to "câu lạc bộ" *club*.

We also built a loanword dictionary (2,821 entries), which comprised pairs of loanword and the corresponding Vietnamese words, e.g. "London Luân-đôn" [lʷɤ̆n-1 ɓon-1], "Ronaldo Rô-nan-đô" [ɗo-1 nan-1 ɗo-1]. The remaining cases, whose full text could not be found by any explicit expansion rules or in any dictionaries, were expanded to a list of words for each letter or character (i.e. a character sequence), e.g. "WTO" was expanded to "vê-kép tê ô" [ve-1 kep-5b te-1 o-1], "NT320" was expanded to "nờ tê ba hai không" [nɤ-2 te-1 ɓa-1 haj-1 xoŋ͡m-1].

3.3.6 Text transcription

After the raw text was cleaned and normalized, syllables in each sentence were transcribed using tonophones as a speech unit. As aforesaid in Section 2.7 in Chapter 2, the PRO-SYLDIC dictionary was constructed to cover all Vietnamese pronounceable syllables with tones. Its entries included both orthographic and transcript form using tonophones. Hence, the PRO-SYLDIC dictionary was used for this text transcription task.

This dictionary covered not only meaningful syllables in Vietnamese, such as "hoa" [hʷˡa1-1] *(flower)*, "cua" [kuə1-1] *(crab)*, "qua" [kʷˡa1-1] *(pass away)*; but also nonexistent but pronounceable syllables such as the loanword "boa" [ɓʷˡa1-1] in 'tiền boa" from French 'pourboire' *(tip)* or the newly appeared orthography syllable "iêu" instead of "yêu" [iə1w1-1] *(*to love).

Since the sources of the raw text were mostly pulled from sources on the Web, they were in different representations. For ease of use with a common format, each source was finally stored in a text file with the following structure: (i) each sentence was in one line, (ii) each line had four columns separated by the special symbol " ˜ ": sentence code ˜ original text ˜ normalized text ˜ transcribed text.

3.4 Phonemic distribution

As presented, the raw text could be considered as a phonetically-rich and balanced source, and might represent for the Vietnamese language in terms of phonetic distribution due to its variety and its large size. This section presents a new and important speech unit in our work, a di-tonophone – an adjacent pair of tonophones. "Theoretical" speech units, which constructed from MEA-SYLDIC – a dictionary of meaningful syllables, are also presented, while "real" speech units were extracted from the raw text.

3.4.1 Di-tonophone

In continuous speech, units were produced with and affected by the preceding and succeeding ones. To cover the transition between two phones, di-phones is usually used in speech synthesis. Using di-phones as a base speech unit, the pronunciation of each phone varies based on the surrounding phones. As a result, di-phones are usually analyzed and play an importance role in corpus design.

As presented in Section 2.6 in Chapter 2, a new speech unit – tonophone – was proposed as an allophone regarding the lexical tone of the bearing syllable. To build tonophones, all allophones in rhymes were adhered to the lexical tone, while the initial consonant did not have to combine any information of the tone. "Tonophones" were used for emphasizing the role of lexical tones, and reflected their corresponding allophones in tonal contexts.

As aforementioned, due to the importance of Vietnamese lexical tones, a new speech unit concerning the lexical tone of the bearing – a "tonophone" was proposed. To build tonophones, all allophones in rhymes were adhered to the lexical tone, while the initial consonant did not have to combine any information of the tone. "Tonophones" were used for emphasizing the role of lexical tones, and reflected their corresponding allophones in tonal contexts. The tonophone set of the Vietnamese language was constructed with 207 units (cf. Section 2.6 in Chapter 2).

In the corpus design, we use a "di-tonophone" as a basic speech unit, which can be defined as an adjacent pair of tonophones. It appears that both phonemic and tonal contexts can be "modeled" in the "di-tonophones". For instance, in the sentence "Trời đẹp quá!" [tɕɤ2j2 ɗɛ6bp6b k^{w5a}a5a] *(What a beautiful day!)*, consuming that there are two empty phones (#) at the beginning and at the end of the sentence, the following di-tonophones were found: [#-tɕ], [tɕ-ɤ2], [ɤ2-j2], [j2-ɗ], [ɗ-ɛ6b], [ɛ6b-p6b], [p6b-k], [k-^{w5a}], [^{w5a}-a5a], and [a5a-#].

3.4.2 Theoretical speech unit sets

As presented in Section 2.7 in Chapter 2, the PRO-SYLDIC dictionary included all Vietnamese pronounceable syllables with tones although many of them does not exist in the language. These entries were useful for a number of loanwords or newly appeared syllables. We assumed that we could build the theoretical di-tonophone set for our work based on a dictionary that included meaningful (i.e. existent) Vietnamese syllables.

MEA-SYLDIC – a MEAningful SYLlable DICtionary. A preliminary analysis was done on the VCL dictionary (cf. Section 3.2) and the raw text. A total of 7,043 meaningful distinct orthographic syllables (and 5,792 distinct transcriptions) were found in the VCL dictionary. Other sources of the raw text provided more loanwords that did not exist in the language, hence a total of 7,355 meaningful distinct orthographic syllables were constructed for a new dictionary – MEA-SYLDIC. The entries of this dictionary included meaningful

hence existent Vietnamese syllables in pairs: 7,355 distinct orthographies corresponding to 6,074 distinct transcriptions. This dictionary was used for building speech unit sets (e.g. tono-phone set, di-tonophone set) and their distributions, which can be considered a reference for the corpus design.

Theoretical unit sets. Theoretical di-phone/di-tonophone set in this work was built based on the MEA-SYLDIC dictionary. Each syllable in the dictionary was combined in pairs with others, e.g. $[a_1a_2a_3]$–$[b_1b_2]$, for generating the theoretical speech units. Before the first phone of the left syllable $[a_1a_2a_3]$ and after the last phone of the right syllable $[b_1b_2]$, we considered an empty phone [#] starting or ending an utterance. For the syllable pair $[a_1a_2a_3]$–$[b_1b_2]$, the following di-phones were: [#-a_1], [a_1-a_2], [a_2-a_3], [a_3-b_1], [b_1-b_2], [b_2-#].

For instance, for the syllable pair "gần – quên" [ɣɤ̆2n2] – [kw1e1n1] *(nearly – forget)*:

- the di-phones were [#-ɣ], [ɣ-ɤ̆], [ɤ̆-n], [n-k], [k-w], [w-e], [e-n], [n-#]
- the di-tonophones were [#-ɣ], [ɣ-ɤ̆2], [ɤ̆2-n2], [n2-k], [k-w1], [w1-e1], [e1-n1], [n1-#]

Following the above method to build di-phones/di-tonophones from the dictionary, there were 1,139 theoretical di-phones while 18,507 theoretical di-tonophones were extracted.

3.4.3 Real speech unit sets

Since the di-tonophone set was automatically generated from all the combinations of syllable pairs using the MEA-SYLDIC dictionary, many theoretical ones did not exist in the raw text (i.e. the Vietnamese real text). As presented, we assumed that the raw text could represent the language in terms of phonetical richness and balance. Therefore, in this work, the speech unit sets and the phonemic distribution of the raw text were considered as real ones and could be used a reference for the corpus design process.

Table 3.2 presents the unit numbers of different sets in theory (using the MEA-SYLDIC dictionary) and the raw text (real Vietnamese texts). Since a number of syllable combinations (or with the empty phone #) did not exist in the language, the number of theoretical di-tonophones (by dictionary) was nearly twice the one in the raw text. Other speech units in the raw text had the same numbers as in the dictionary. The total number of transcribed syllables was 6,074[2].

3.4.4 Distribution of speech units

Based on the above building of speech unit sets, we calculated the distribution of units for further tasks. Table 3.3 lists the top 9 frequent (p1-p5) and rare (r5-r1) phones, tonophones, di-phones and di-tonophones in the raw text. In the raw text, [ɣ] was the rarest phone while the rarest tonophone was [ɔ̆] with the broken tone 4. The phones [k͡p ɔ̆ ɛ ɛ̆] were also the rarest ones. The broken tone 4, the curve tone 3 and the drop tones (6a, 6b) seemed to be rare, especially combining with [uə ɔ̆ e]. The phone [a] was the most frequent and the [j] was the fifth common in the raw text. Disregarding the lexical tones, the phones [k n t] (and also [m ŋ tɕ]), which can be both initial and final consonants, were also the most popular ones.

As for tonophones, since the lexical tones were adhered only to the rhymes, the initial consonants [k tɕ ɗ t] and some other initial ones (e.g. [v h z t^h m l ɓ s n]) were the most common in the raw text. The vowel [a] with the level tone was the fifth frequent in the raw text. The more detail distribution showed that the level tone 1 and the falling tone seemed

2. The number of orthographic syllables was 7,355.

Table 3.2 – Number of speech units in theory and in the raw text

#	Factor	Dictionary (theory)	Raw text (real)
1	Number of sentences	-	323,934
2	Number of distinct phones	48	48
3	Number of distinct tonophones	207	207
4	Number of phones	-	28,329,368
5	Number of distinct initials/rhymes	674	674
6	Number of initials/rhymes	-	20,400,713
7	Number of distinct di-phones	1,139	1,139
8	Number of di-phones	-	28,653,194
9	Number of distinct di-tonophones	18,507	10,339
10	Number of distinct syllables	6,074	6,074
11	Number of syllables	-	10,377,903

Table 3.3 – Distribution of top 9 frequent (p1-9) and rare (r9-1) speech units of the raw text

#	Phone	Freq.	Tono-phone	Freq.	di-phone	Freq.	Ditono-phone	Freq.
p1	a	2,367,636	k	1,120,216	a-j	406,534	o1-ŋm͡1	210,693
p2	k	1,831,035	tɕ	978,603	o-ŋm͡	363,465	ă1-m1	196,935
p3	n	1,828,931	ɗ	888,552	iə-n	339,192	v-a2	187,790
p4	t	1,396,160	t	795,718	a-n	318,623	a5b-k5b	167,220
p5	j	1,311,072	a1	714,008	i-j	291,065	a1-j1	157,150
p6	m	1,131,790	h	622,791	ɤ̆-n	280,005	k-a5b	121,863
p7	o	1,003,581	t^h	565,268	a-k	267,868	h-a1	121,820
p8	ŋ	980,619	m	551,041	^{w}-a	262,602	a6a-j6a	121,506
p9	i	950,371	a2	544,490	a-m	257,528	n-ă1	115,870
r9	f	292,890	ɤ4	5,305	ɤ-ŏ	3	j5a-^{w2}	1
r8	z	271,039	ă4	4,707	ɛ-ŏ	3	j3-ɯə1	1
r7	x	259,775	ɤ5b	4,566	uə-uə	3	n6a-ɤ̆2	1
r6	p	258,420	ɛ6b	3,968	iə-ɯ	2	j6a-a6a	1
r5	ĕ	243,998	e6b	2,315	iə-ɔ	2	t5b-ɛ2	1
r4	ɛ	218,326	uə6a	1,946	u-ŏ	2	k͡p5b-i2	1
r3	ŏ	193,806	ŏ3	1,690	ɯə-ŏ	2	k͡p5b-i3	1
r2	k͡p	109,232	uə4	612	uə-ŏ	1	n5a-ɛ6a	1
r1	ɣ	76,905	ŏ4	229	iə-ŏ	1	m2-uə3	1

to be the most popular ones, especially combined with popular phones (e.g. [a n j ŋm͡ m ă N o i]). The final consonants [p t k] were also common (with the rising tone 5b or the drop tone 6b).

The most frequent di-tonophones in the raw text were popular rhymes, such as “ông” [o1-ŋm͡1], “am” [ă1-m1], “ác” [a5b-k5b], “ai” [a1-j1]. The di-tonophone “và” [v-a2] was the third common. Despite regardlessness of the lexical tones, the di-phones [a-j] and [o-ŋm͡] were still the most frequent. Some combinations of two nucleus, vowels or diphthongs (especially the rare ones), were rare such as [iə-ŏ], [uə-ŏ], [ɯə-ŏ]. The unusual combinations of lexical

tones provided rare di-tonophones, such as [m2-uə3], [n5a-ɛ6a], [k͡p5b-i3], etc.

There were a number of di-tonophones with small frequencies. Table 3.4 shows the numbers of di-phones and di-tonophones having the frequency from one to six. Nearly 1,200 di-tonophones appeared only once and 615 twice in the raw text. The numbers of di-tonophones with the frequency of three to six ranged from 381 down to 148. Only 2 to 4 di-phones had small frequencies.

Table 3.4 – Number of di-phones/di-tonophones having small frequencies

Frequency	**di-phone #**	**Di-tonophone #**
Once	2	1,199
Twice	4	615
Three times	6	381
Four times	2	257
Five times	2	216
Six times	3	148

3.5 Corpus design

Due to the simplicity and effectiveness, the greedy algorithm was adopted to search the best candidate among the subset of the raw data. This section describes the whole corpus design as a number of iterations of selection process, whose output was the best candidate in terms of phonetic-richness and -balance at the current state of the uncovered units and their distributions. The selection process stopped when an expected constraint reached.

Three corpora were then constructed by our proposed design process with different speech units and targets: (i) SAME: a new corpus with the same size as the old corpus VNSP in terms of syllable number and using di-tonophones as speech units, (ii) VSYL: a new corpus with 100% syllable coverage (i.e. complete syllable coverage), and (iii) VDTS: a new corpus, which had 100% di-tonophone coverage (i.e. complete di-tonophone coverage). The purpose of first corpus was to examine the performance of the algorithm by comparing the distribution of different speech units of the old corpus and the new corpus with the same size. The second one was designed for the non-uniformed unit selection in which syllables can be used as the best speech unit in terms of both quality and system/corpus size. The last one was designed for our TTS system, in which tonophones were used as speech units. We believed that with the design of 100% di-tonophones, the transitions between any two tonophones were completely covered, hence the quality would be much more improved.

3.5.1 Design process

The speech unit sets and their distributions of the raw text (cf. Section 3.4) were used in the selection process as well as weighting candidate sentences. Figure 3.2 illustrates the process of corpus design, which includes a number of selection iterations to build a target corpus with an expected constraint.

The input data of the selection process were: (i) e: the expected coverage (e.g. 100% for full coverage) or condition (e.g. max size of the target corpus), (ii) R: the **r**aw text including transcribed sentences, and (iii) U: the **u**ncovered speech unit set with frequency. The initial value of U was the whole speech unit set with frequency of the raw text as mentioned in

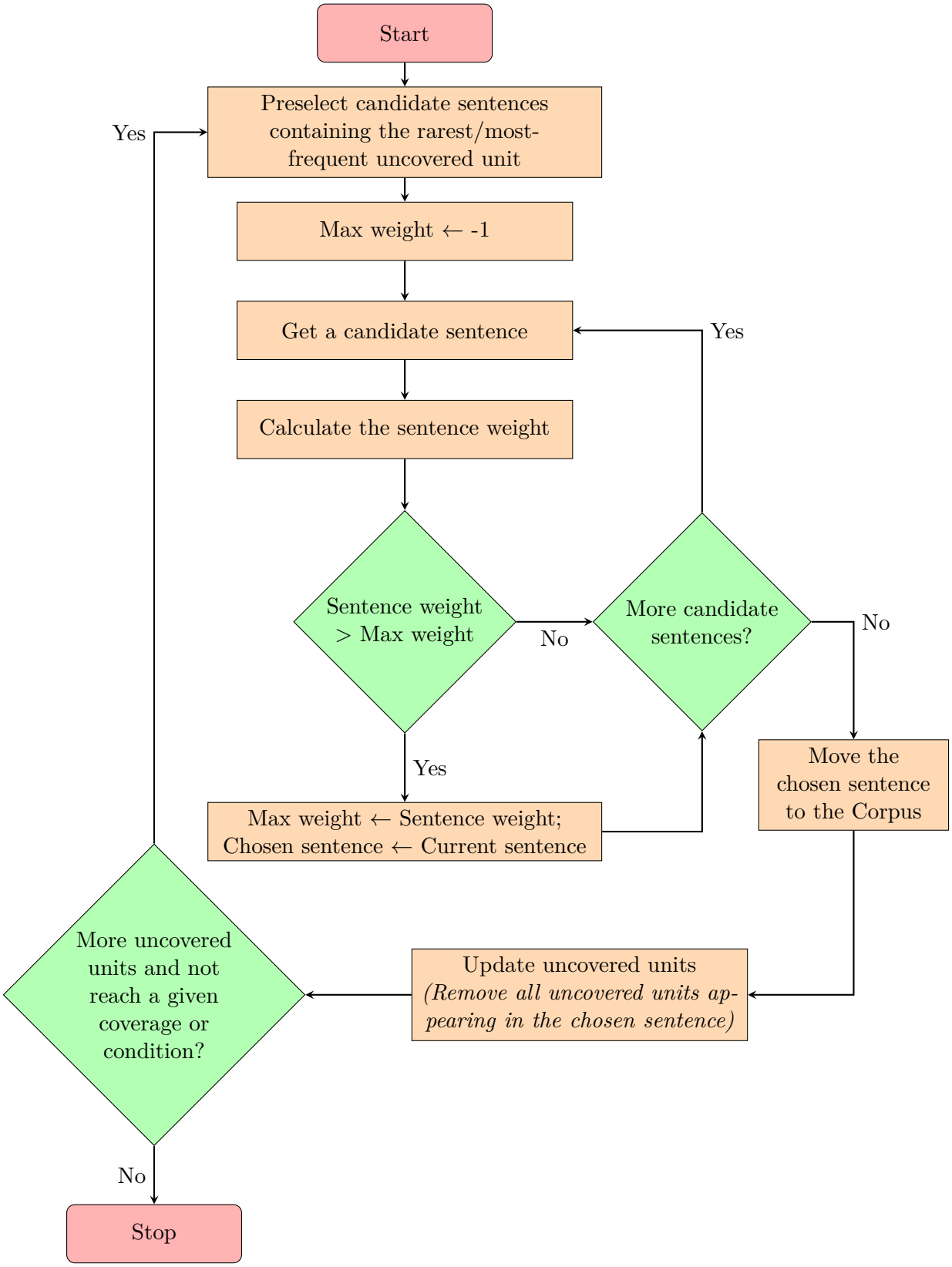


Figure 3.2 – Corpus design: repetitions of selection processes.

the previous section. The output of each selection process was a chosen sentence, which was considered as the best candidate in terms of phonetic-richness and -balance.

The criteria for the whole design process was to have: (i) the highest coverage possible of a given speech unit (e.g. di-phone) with (ii) the smallest corpus possible (i.e. a smallest number possible of chosen sentences). The output of the design was a set of chosen sentences T–the **t**arget corpus, with its distribution.

First, from the raw text R, sentences that included the rarest speech unit of the uncovered unit set U were chosen as a set of n **c**andidate sentences $C = S_1, S_2, \ldots, S_n$. We assumed that in the phonetically-rich and -balanced raw text, sentences containing rare speech units might include more common ones, but the possibility of vice versa was much smaller. However, if the coverage of the target unit is not 100% (e.g. 70%), the most frequent unit may be chosen to optimize the corpus size. The corpus design may need to be performed twice to find out the best solution for choosing the rarest or most frequent uncovered speech unit.

The weight of each candidate sentence S_i in C was then calculated to choose the most phonetically rich sentence among n sentences $C = S_1, S_2, \ldots, S_n$. At the time of selection, the richness of one sentence could be represented by its number of uncovered distinct units. However, in general, the longer sentences had more speech units, hence more distinct ones. Therefore, the weight of one sentence was normalized by its number of all distinct units, as illustrated in Equation 3.1. The sentence with the maximum weight S_c was considered as the richest one and moved to the target corpus T.

$$Weight(S_i) = \frac{N_{ui}}{N_{ai}} \tag{3.1}$$

where

- S_i: The sentence i in n candidate sentences C
- N_{ui}: Number of **u**ncovered distinct units appearing in the sentence S_i
- N_{ai}: Number of **a**ll distinct units in the sentence S_i

After having chosen the best candidate sentence S_{cj}, the uncovered speech unit set with frequency U was updated. All distinct speech units in S_{cj} were removed from U. If this uncovered unit set had more elements and the given coverage/condition was not reached, the selection process was repeated to build a new set of candidate sentences C and so on. This process stopped when U was empty or the target corpus T including m sentences $S_{c1}, S_{c2}, \ldots, S_{cm}$ satisfied the given condition or coverage c. If the U was empty at the end of the selection, T had a full coverage (100%), meaning that it covered the whole speech unit set of the raw text.

3.5.2 The constraint of size

To examine the performance of the proposed design process, we carried out a design that considered di-tonophones as speech units with a constraint of the target corpus size. The output was a new text corpus with a similar size (i.e 24,164 number of phones) to the old one – VNSP (630 sentences, 8,930 syllables). Since the bounded size of the target corpus was small and the number of di-tonophones appearing once in the raw data was considerable (i.e. 1,199 times), we assumed that in the selection process, sentences containing the most frequent speech unit should be chosen as candidates for weight calculation. If the rarest speech unit were considered first, sentences including those single-occurrence di-tonophones would

Table 3.5 – New corpora designed with the same size as the old one VNSP. SAME: candidate sentences containing the most frequent uncovered unit, SAME-B: candidate sentences containing the rarest one

#	Factor	VNSP (old)	SAME (new)	SAME-B (new)
1	Number of sentences	630	983	334
2	**Mean length (syllables/sentence)**	**17.1**	**9.6**	**27.1**
3	Coverage of phones	100.0%	100.0%	100.0%
4	Coverage of tonophones	95.1%	100.0%	100.0%
5	**Number of phones**	**24,164**	**24,117**	**24,021**
6	Coverage of initials/rhymes	65.3%	84.0%	73.4%
7	Number of initials/rhymes	17,504	17,778	17,433
8	Coverage of di-phones	74.5%	91.7%	88.5%
9	**Coverage of di-tonophones**	**29.6%**	**52.4%**	**37.2%**
10	Number of di-phones	24,686	25,100	24,355
11	Coverage of syllables	24.8%	44.6%	33.0%
12	**Number of syllables**	**8,930**	**9,478**	**9,048**

have been the unique candidates and hence have been chosen. This would have reduced the coverage of the target corpus.

In fact, in order to confirm our assumption, we did the corpus design twice with two different preselection conditions of candidate sentences: (i) the rarest speech unit, and (ii) the most frequent one. The Table 3.5 provides the total numbers and coverages of different speech units of the two new corpora "SAME", "SAME-B" and the old one "VNSP". The selection process was iterated while the syllable number of the target corpus ("SAME" or "SAME-B") did not exceed the size of "VNSP" in terms of phones (i.e. 24,164 phones). Since one sentence was chosen in each iteration, the phone numbers of the two new corpora were slightly smaller than that of the old one (0.2-0.6%), while the syllable numbers were a bit larger (1.3-6.1%).

The coverage of the new corpus "SAME" was much higher than the old one. There was no or small difference of the phone or tonophone coverages, yet wide gaps (about 17-22%) of the other unit coverages between these two corpora. The di-tonophone coverage of the new corpus reaches 52.4%, while that of the old one was only 29.6%. The coverage of the "SAME-B" corpus was rather higher than the old one, only about 7-14%.

3.5.3 Full coverage of syllables and di-tonophones

The corpus of high-quality TTS systems should have a good coverage of speech units. With unlimited vocabulary, some TTS systems even require a corpus with a full coverage (100%) of the target speech unit.

For instance, in a non-uniform unit selection TTS system for Vietnamese – HoaSung (Do et al., 2011), due to a small corpus (VNSP – 630 sentences), the half-syllable corpus of Tran (2007b) was used when there were lack of syllables or above syllables when searching units. However, the appearance of half-syllable units sometimes degraded the quality of the synthetic speech at discontinuous points. Moreover, even with the half-syllable corpus, there was still lack of many instances of that unit leading a failure of the synthesis process or an non-intelligible speech. As a result, there was a need to design a corpus having a complete

syllable coverage (i.e. 100% syllable coverage) to ensure the stability and the quality of the synthetic voice.

For VTED, a Vietnamese HMM-based TTS system (Nguyen et al., 2013b, 2014a,b), tonophones were used as a speech unit for training and synthesis. This system needed a corpus with a good coverage in both phonemic and tonal contexts. To completely record all the transitions between any two tonophones, it was necessary to design a corpus with 100% di-tonophone coverage.

Based on the requirement of the two above system, the proposed design process was performed for two corpora for different speech units: (i) VSYL (Vietnamese SYLlable speech) corpus with 100% syllable coverage, and (ii) VDTS (Vietnamese DiTonophone Speech) corpus with 100% di-tonophone coverage. As presented in Section 3.4, the number of di-tonophones appearing once in the raw data was considerable (1,199 times). It means that in order to cover the complete di-tonophone set, all sentences containing these once-occurence di-tonophones must be included in the target corpus. Hence, the rarest uncovered speech unit was considered in the preselection of candidate sentences [3]. A similar process was run for the VSYL corpus.

3.5.4 VDTS corpus

As presented above, the VDTS corpus was designed with a full coverage of di-tonophones. Obviously, the VDTS corpus had 100% coverage of phones, tonophones, and di-phones. Its coverages of initial/rhymes and syllables were 95.1% and 70.2% respectively. VDTS was the target corpus that we used for our TTS system as a new training corpus. We expected that using a corpus with a complete di-tonophone coverage, the quality of the synthetic speech would be much improved since the transitions between any two tonophones were completely recorded.

Table 3.6 – VSYL – the corpus with a complete syllable coverage, and VDTS – the corpus with a complete di-tonophone coverage

#	Factor	**VSYL corpus** (100% syllable)	**VDTS corpus** (100% di-tonophone)
1	Number of sentences	2,297	3,947
2	**Mean length (syllables/sentence)**	**14.4**	**21.5**
3	Coverage of phones	100.0%	100.0%
4	Coverage of tonophones	100.0%	100.0%
5	**Number of phones**	**90,219**	**223,806**
6	Coverage of initials/rhymes	100.0%	95.1%
7	Number of initials/rhymes	59,978	161,897
8	Coverage of di-phones	92.3%	100.0%
9	**Coverage of di-tonophones**	**57.0%**	**100.0%**
10	Number of di-phones	92,516	227,753
11	Coverage of syllables	100.0%	70.2%
12	**Number of syllables**	**33,033**	**84,769**

Table 3.6 shows the results of these two corpora that had full coverages of syllables/di-tonophones. The VSYL corpus has 100% coverage of phones/tonophones, syllables and ini-

3. We run the design process twice, and the result was: the corpus designed with the most frequent uncovered unit for the preselection of candidate sentences had bigger size than that designed with the rarest one

tial/rhymes. However, its di-tonophone coverage was only about 57.0%. The VSYL corpus had nearly 2,300 sentences while there were nearly 4,000 sentences in the VDTS corpus (100% di-tonophone coverage). The numbers of phones of VDTS was about three times that of the VSYL one. The VDTS corpus had a good syllable coverage (70.2%) and initials/rhymes coverage (95.1%).

3.6 Corpus recording

In this work, beside more than 4,700 sentences including the VDTS corpus (the new training corpus for our TTS system) and some sentences for special purposes or the evaluation phase; the text content of the VNSP corpus (the old one from the previous studies) was also recorded for comparison. A total of 5,338 sentences were recorded at LIMSI, France by a female non-professional native speaker from Hanoi, aged 31 (named Nguyen Thi Thu Trang; Thu-Trang for short). The speaker had left Hanoi 2 months at the time of the recording. Although she was not a professional speaker, she had a natural and quite pleasant reading style with a suitable prosodic representation. She was a lecturer hence she was able to maintain the voice quality during the recording session.

3.6.1 Recording environment

The recordings took place in a studio at the LIMSI-CNRS Laboratory, Orsay, France. The recording studio included a soundproof vocal booth and a control station.

In the soundproof booth, there were the following equipments: (i) a condenser microphone with an omnidirectional polar pattern, (ii) a Glottal Enterprises EG2 glottograph, (iii) an iPad allowing the speaker to access text content of sentences, and (iv) a loudspeaker. The speaker position was controlled in the beginning of each session by measuring a fixed distance (about 30 cm) from the mouth of the speaker to the microphone. A round anti-pop filter was located in front of the microphone.

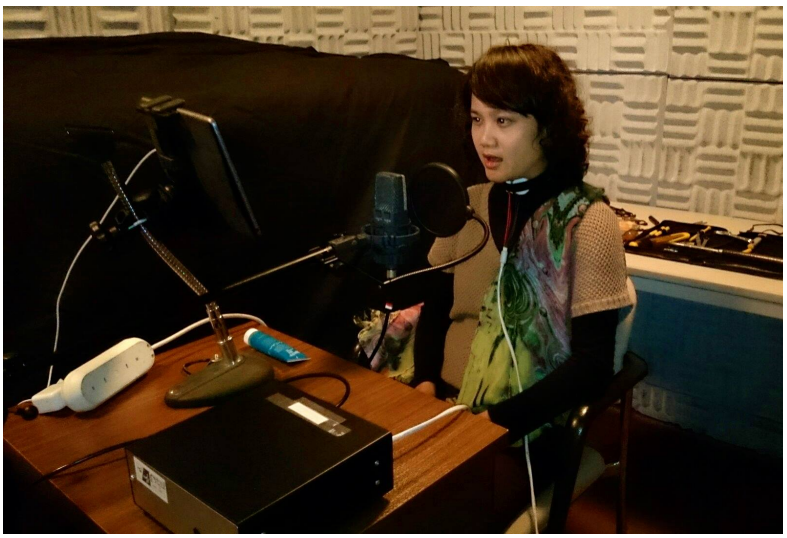


Figure 3.3 – Soundproof vocal booth. The iPad screen was put in a suitable and straight position for the speaker. The anti-pop filter was in front of the microphone.

The control station, operated by a recording supervisor, included: (i) a computer (iMac

21.5-inch, Mid 2011), (ii) a high-quality sound card (RME Fireface 400), (iii) and a headphone. The audio and EGG [4] signals were recorded directly into the computer using the software Pro Tools 10 [5] through the sound card at a sampling rate of 48,000 Hz and 24-bit quantization. They were eventually converted to 48,000 Hz, 16-bit PCM files. Due to private reasons, only audio files were used in this work.

3.6.2 Quality control

The recordings were done in one-hour sessions with a 5 minutes interval every half hour so that the speaker throat could have an enough relax to ensure the recorded speech quality. The speaker recorded from two or four (rarely) sessions per day. Each recording session produced, on average, 200 utterances, hence about 17 minutes of recorded speech. About 7.7 speech hours (462 minutes) corresponding to 5,338 utterances were recorded; hence 27 recordings sessions were conducted.

The speech quality was controlled during and after the recording sessions. To facilitate the quality control and the further analysis, the sentence-by-sentence recordings were conducted. To provide references of voice level and quality, in the beginning of each session the speaker listened to some recordings of the previous sessions. The audio feedback and the supervisor instructions were routed to the speaker's loudspeaker and supervisor's headphones. They could also communicate to each other by gestures through a glass window between the booth and the control station.

Supervision during the recording sessions. The supervisor was a Vietnamese native speaker. He was trained to use the recording software as well as other constraints for a good-recorded speech. The supervisor operated the recording software to monitor the sound level, to start and stop the recorder and to erase the error utterances. It was also the responsibility of the supervisor to verify if the speaker was producing the right sound level, either by moving away from the predefined position or by starting to become tired. The supervisor also had to check if the speaker read all the words in the sentence with the proper pronunciation using an adequate rhythm and intonation. When there was any problem, he stopped the recording and explained the issues to the speaker. They may listen to that sound again to clarify the problems and confirm the right way to read that sentence. The supervisor canceled the previous one to override with a new utterance. One of the most challenging sessions involved reading loan words or rare words, and long sentences.

Verification after the recording sessions. To reduce errors in the speech corpus as much as possible, the audio files were periodically checked after several sessions. The speech rate, voice level and quality were first compared between the unchecked sessions and checked sessions. This could detect if there were any change or errors in the recording conditions for different sessions.

The errors might be caused by the supervisor when he forgot canceling the error utterances, started too late or stopped too early. This might lead to the audio files having too short margin pauses or the speech being improperly cut. The utterances with errors were discarded and the sentences were re-scheduled for future recording sessions.

4. ElectroGlottoGraph, also called laryngograph
5. The industry-standard audio production platform: http://www.avid.com/US/products/family/pro-tools

3.7 Corpus preprocessing

Corpus preprocessing had to be done to build a "clean" and annotated speech corpus for TTS systems. This section presents the three major post-recording tasks: (i) normalizing the beginning and ending pauses, (ii) labeling the continuous speech according to the phonetic transcription, and (iii) processing the wrong labeling of breath noises.

3.7.1 Normalizing margin pauses

Each recording file corresponding to one sentence was named increasingly by an incremental value of "1". The first step was to rename these recording files to new names corresponding to sentence codes (cf. Section 3.3). Each file was trimmed to margin of at-most-200ms pauses in the beginning and at the end. These tasks were automatically done using a Praat [6] script.

Since the recordings and quality control were made by humans, there were still some error audio files. For instance, the verbal content of some utterances and the corresponding text content mismatched, such as lacking or redundant syllables, or even wrong syllables. The margin's pauses of some audio files were too short (e.g. <100ms), or the speech signals were improperly cut in some files. Some utterances had unexpected noises because of the carelessness during the recordings. Utterances with unsolvable errors (e.g. deeply cutting speech, "strong" noises) were removed from the corpus. Some text files were modified to suit the verbal content of the respective audio files.

3.7.2 Automatic labeling

With the increase in size of speech databases, manual phonemic segmentation and labeling of every utterance became unfeasible. Thus, automatic labeling was one important task to build an annotated corpus for TTS systems.

In this work, the updated text files were first transcribed into phonemic sequences using our G2P conversion module. The speech corpus was then segmented and force-aligned with the orthographic transcriptions by the EHMM labeler [7] (Anumanchipalli et al., 2011) since it was well tuned to automatic synthesis labeling. Given a phonetic transcript of a speech and the waveform, the EHMM tool automatically finds the alignments between the speech and the transcript at the phone level. The quality of the labeling often depends on the amount of data trained making it appropriate for segmentation of large speech databases.

Actually, MaryTTS was adopted as the platform for developing our TTS system. Therefore, we used the supporting tools and default configurations of EHMM from MaryTTS for this labeling. In the EHMM tool, continuous models with one Gaussian per state, left-to-right models with no skip state and context-independent models trained with 13 MFCCs are used to get force-aligned labels. It supports modeling of short, long, and optional pauses, which produces better alignment of speech segments. More resolution can be supported with a frame shift of 5 milliseconds resulting in sharper boundaries (Schröder et al., 2008).

EHMM uses a context-independent acoustic model, i.e. an acoustic-phonetic unit set of the target language, as context dependent models tend to blur the label boundaries. This acoustic unit set is intended to represent every speech segment that is clearly bounded from an acoustic point of view, as well as every speech segment that is phonetically significant, even if it is not clearly bounded. In other words, it is used to model and to identify clear acoustic events that an expert phonetician would mark as boundaries in a manual segmentation session. The

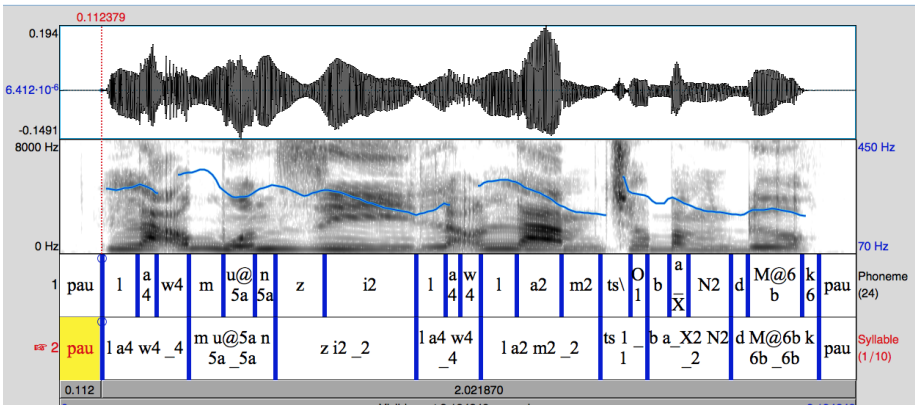
6. Praat: doing phonetics by computer: http://www.fon.hum.uva.nl/praat/
7. A labeler included in the festvox project: http://festvox.org/, from festvox-2.1

acoustic-phonetic tonophone set of the Vietnamese presented in Section 2.6 was used as an input for the EHMM labeler.

This software extracted cepstral coefficients from the wave files and trained HMMs using the Baum Welch algorithm to determine the phone boundaries. The outputs of the EHMM labeler were HTK style labels. The first column contained the phone end times in millisecond, the last column the phone symbol. We had to convert them into the TextGrid format for verification.

The EHMM labeler had been shown to be very reliable, and could nicely deal with pause insertion. However, its main drawback was the speed. For the old corpus containing 630 sentences, it took several hours for labeling using a "iMac 21.5-inch, Mid 2011". For the new corpus VDTS containing nearly 4,000 sentences, we had to spent one day and a half.

(a)



(b)

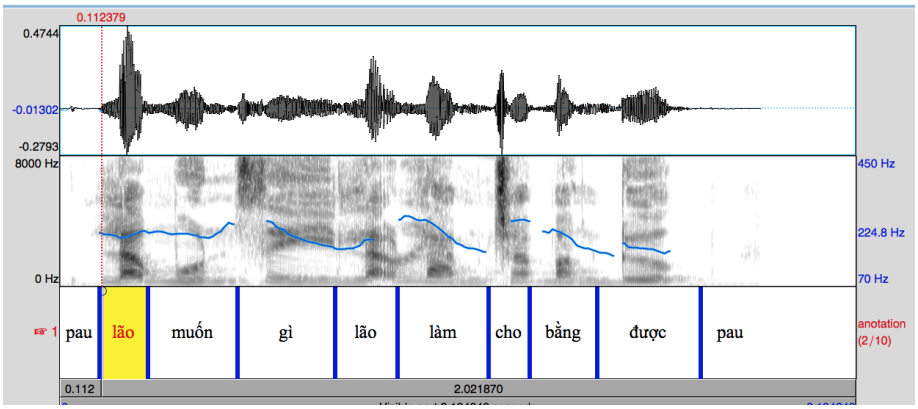


Figure 3.4 – An example of transcription files (TextGrid) for sentence "Lão muốn gì lão làm cho bằng được" [law-4 muən-5a zi-2 law-4 lam-2 tɕɔ-1 ɓăŋ-2 ɗɯək-6b] in (a) the old speech corpus by a broadcaster (manual labeled) and (b) the new speech corpus by ThuTrang (automatic labeled).

Semi-automatic correction of breath noise labeling. A primarily perceptual evaluation showed that the synthetic voice trained with the first result of the annotated corpus sounded discontinuous at some transitions between syllables. Some observations were done on the labeled speech of the training corpus for discovering the problems. Although EHMM could deal with pause insertion, but it often failed to predict the pause appearance or pause duration

in the speech corpus since the speaker mainly produced breath noises instead of silence pauses. A number of breath noises were confused with the adjacent segments. In some cases, a part of the breath noise could also be wrongly labeled, where a pause were labeled yet with a much smaller duration.

These wrongly-labeled breath noises were automatically identified based on the duration limits of phone types (vowels/consonants, short/long tones) in the corpus. The durations of the phones including the breath noises were adjusted to the sum of their means and standard deviations. We also had to do some manual corrections for some special cases, i.e. too-small pauses, single-vowel syllables, syllables including long vowels or semi-vowels, etc. More information about this issue and the correction process are described in Section B.1 in Appendix B.

As a result, the new corpus were segmented and labeled at phoneme-level using tonophones. For further analysis, tonophones in the new corpus were then grouped to syllables and perceived pauses in a different tier, as illustrated in Figure 3.4a. Whereas, the old speech corpus VNSP was manually labeled, time-aligned at the syllable level in graphemes, and annotated for perceived pauses, as shown in Figure 3.4b. This was the existing result of the annotated corpus VNSP from the previous works (Tran, 2007b) (Nguyen et al., 2013b, 2014a,b).

3.7.3 The VDTS speech corpus

Due to few errors in automatic sentence segmentation of the raw text as well as in recording, the utterance number of the VDTS speech corpus (i.e. 3,947 utterances) was slightly different from the sentence number of the designed VDTS text corpus. Regarding pauses inside utterances, a total of about 6.4 hours (i.e. 384 minutes) of speech were obtained. As aforesaid, this new corpus was recorded in France by a female non-professional speaker Thu-Trang from Hanoi at 48 kHz, 24 bps and eventually converted to 48 kHz, 16 bps.

There were totally 8,506 perceived pauses inside utterances of the speech corpus VDTS. In average, the speaker produced a perceived pause every nine syllables. The speech rate of VDTS was about 9.6 phonemes/s, or 3.6 syllables/s.

A total of 630 sentences in the existing text corpus VNSP, as presented, were also recorded by the speaker Thu-Trang for comparison. Section B.2 provides some comparisons between the two speech corpora with a similar text content but recorded by different speakers and recording condition. The old corpus was recorded in Vietnam by a female broadcaster from Hanoi at 16 kHz and 16 bps (hence called VNSP-Broadcaster); meanwhile the new one was recorded in France by a female non-professional speaker Thu-Trang from Hanoi at 48 kHz, 24 bps and eventually converted to 48 kHz, 16 bps (hence called VNSP-ThuTrang).

3.8 Conclusion

A speech corpus can be considered a phonetically rich one if it contains all the phones and has a good coverage of other speech units (e.g. di-phones, triphones), which can capture the transitions between two phones. A phonetically balanced corpus maintains the phonetic distribution of the language. In this work, a vast bank of raw text, which was crawled from various sources (i.e. e-newspapers, e-stories, Vietnamese word dictionary, etc.), was considered a phonetically-rich and -balanced resource and a reference for the design process. This resource might represent for the Vietnamese language in terms of phonetic distribution.

Since the raw text was huge and was collected from various sources, there was a need to

pre-process to make it to be suitable for the design process with five main tasks: (i) Sentence segmentation, (ii) Tokenization, (iii) Text cleaning, (iv) Text normalization, and (v) Text transcription. Texts were first segmented into sentences, and then tokenized into syllables or Non-Standard Words (NSWs – which cannot be directly transcribed to phonemes, e.g. numbers, dates, abbreviations, currency). Sentences having more than 70 syllables or containing unreadable characters (e.g. control ones) were removed. The next task for "cleaned" sentences were was text normalization, in which NSWs were then processed and expanded to speakable syllables. The normalized text was finally transcribed into tonophones to provide a suitable input for next steps.

Since Vietnamese is a tonal language, our work targeted to design a phonetically-rich and -balanced corpus in both phonemic and tonal context. Hence, two proposed speech units used for designing corpora for Vietnamese TTS systems were: (i) a "tonophone": a phone regarding the lexical tone of the bearing syllable, and (ii) a "di-tonophone": an adjacent pair of "tonophones". The di-tonophone set was constructed using a dictionary including meaningful syllables (theoretical), and using the raw text (real). The phonetic distribution of the raw text was calculated for different speech units, including the two new ones.

The whole corpus design included a number of iterations of selection process, whose output was the best candidate in terms of phonetic-richness and -balance at the current state of the uncovered units and their distributions. The selection process could be described as follows. A subset of the huge raw data including the rarest/most frequent uncovered unit was extracted to achieve a set of candidate sentences. Due to the simplicity and effectiveness, the greedy algorithm was adopted to search for the best candidate sentence (with the highest weight) among that subset. The weight of a sentence was the proportion of its uncovered distinct unit number and its total distinct unit number. After each selection, the uncovered unit set was updated, and the selection process was repeated until a constraint (e.g. coverage, condition) was satisfied.

To examine the performance of the proposed design process, we carried out the design that considered di-tonophones as speech units with a constraint of the target corpus size. The output was a new text corpus, "SAME", with a similar size (i.e 24,164 number of phones) as the old one – VNSP. The bounded size of the target corpus was small and the number of di-tonophones appearing once in the raw data was considerable. If sentences containing the rarest speech unit were considered first, sentences including those single-occurrence di-tonophones would have been the unique candidates and hence have been chosen for the target corpus. Therefore, sentences containing the *most frequent speech unit* were chosen as candidates for weight calculation for maximizing the coverage of the target corpus. The results show that with a similar syllable number, the coverage of the new corpus "SAME" was much higher than the old one. There was no or small difference of the phone or tonophone coverages, yet wide gaps (about 17-22%) of the other unit coverages between these two corpora. The di-tonophone coverage of the new corpus reaches 52.4%, while that of the old one was only 29.6%. The VSYL corpus with a complete syllable coverage was also designed for improving the quality of the non-uniformed unit selection speech synthesis.

The target training corpus for our TTS system, the VDTS (Vietnamese DiTonophone Speech) corpus, was designed with a full coverage of di-tonophones since it is necessary to record all the transitions between any two tonophones. Obviously, the VDTS corpus had 100% coverage of phones, tonophones, and di-phones. Its coverages of initial/rhymes and syllables were 95.1% and 70.2% respectively.

A total of 5,338 sentences including the VDTS and VNSP corpora, and some other sentences (called VDTO – Vietnamese Di-Tonophone and Others) for the evaluation phase were

recorded by a female non-professional native speaker from Hanoi, aged 31 (named Thu-Trang). The recordings were conducted in a well-equipped studio including a soundproof vocal booth and a control station at the LIMSI-CNRS Laboratory, Orsay, France. There were 27 one-hour recording sessions to produce nearly 8 speech hours. The speech quality was controlled during the sessions by a supervisor. After several recording sessions, audio files were checked to ensure the global quality and to reduce errors for next sessions.

Utterances in the speech corpus were renamed by sentence codes and pre-processed for our Vietnamese TTS system. They were automatically segmented and force-aligned to build an annotated corpus by the EHMM labeler. However, there existed wrongly-labeled breath noises, which made discontinuous transitions between syllables of the preliminary synthetic voice. These breath noises were hence semi-automatically corrected for a final annotated corpus.

The VDTS speech corpus that was used for training VTED finally contains 3,947 utterances in about 6.4 hours (384 minutes). The speech rate of VDTS was about 9.6 phonemes/s, or 3.6 syllables/s, hence about 25% lower than the previous one recorded by a broadcaster. In average, the speaker Thu-Trang produced a perceived pause every nine syllables, about 16% more pauses than the broadcaster.

Chapter 4

Prosodic phrasing modeling

Contents

4.1 Introduction

In Vietnamese, like in other tonal languages, pitch is used as a part of speech and can change the meaning of a syllable/word. The utterance-level intonation hence not only varies over the course of the sentence, but also interacts with lexical tones.

Trần Đỗ Đạt (Tran, 2007b, Tran and Castelli, 2010) did the analysis on the influence of coarticulation effect and syllable duration on variations of Vietnamese tones in continuous speech. A method for generating F0 contour was then proposed by concatenating tonal contours (tone patterns in accordance with their durations) placed on register contours of syllables composing breath groups in sentences. This work showed a dependence of the global intonation on the lexical tones of constituent syllables.

The tone of the final word or syllable and the intonation-type of the utterance (either declarative or interrogative sentences) have been investigated in a lot of research. The work of Yuan et al. (2002), Zeng et al. (2004) revealed that in Mandarin, the interrogative intonation had a "higher sentence-final melodic curve than the declarative counterpart". However, this difference was complicated because of the interaction of lexical tones and intonation. For instance, the interrogative intonation with a final tone with falling contour often has a falling end (Yuan et al., 2002). In Vietnamese with 8 different tones (only two tones with rising contour), this was much more complex. The study of Vu et al. (2006) confirmed the role of lexical tones in this difference. In the work of Le et al. (2011), from the declarative sentence, a model of F0 contour for yes/no questions without auxiliary verbs was proposed by two main stages: (i), the whole contour was raised by a range of percentages of the F0 mean (normalized register ratio); and (ii) the contour of the final syllable was also raised by a range of percentages of the F0 mean (increasing slope) (Le et al., 2011). However, this work reported that the model did not work well with some particular final syllable tones, e.g. falling and curve tones, due to the small analysis corpus and the absence of investigation on lexical tones. It turns out that the utterance-level intonation poses a constraint on lexical tones. The lack of such a constraint in prosodic modeling may negatively impact the characteristic of syllable tones, hence syllable meanings.

In the HMM-based speech synthesis, speech parameters including spectral (e.g. mfcc) and excitation ones (e.g. F0) are statistically modeled and generated by using context dependent HMMs. Each HMM also has its state-duration distribution to model the temporal structure of speech. As a result, prosodic cues such as F0 or duration can be well learned in both phonemic and tonal context (especially using tonophones as a speech unit, cf. Chapter 2, 3). In other words, the utterance-level intonation can be statistically modeled with a constraint on lexical tones. This considerably increases the naturalness of the synthetic voice. The remaining problem in prosodic analysis is prosodic phrasing; the process of inserting prosodic breaks in an utterance. It includes pause insertion and lower levels of grouping syllables. In an HMM-based TTS system, a pause is considered a phoneme; hence its duration can be modeled. However, the appearance of pauses cannot be predicted by HMMs. Lower phrasing levels above words may not be completely modeled with basic features.

Prosodic phrasing is a crucial step in speech synthesis since other prosodic cues depend on it. The synthetic speech with a better phrasing is more intelligible and natural. Many researchers have been working on prosodic structure generation for Chinese (Chou et al., 1996)(Tao et al., 2003), break modeling (Doukhan et al., 2012) or prosodic structure (Martin, 2010) for French, pause modeling for German (Apel et al., 2004), Russian (Chistikov and Khomitsevich, 2013) or style-specific phrasing (Jokisch et al., 2005)(Parlikar, 2013). They may use rules or machine learning with lexical information (e.g. POS tags) or contextual lengths.

However, to the best of our knowledge, there is no such work on the Vietnamese language. Due to the constraint of intonation with the lexical tones in Vietnamese, it appeared too difficult to disentangle intonation from lexical tones. As a result, in this research, we aimed at prosodic phrasing for the Vietnamese TTS using durational clues alone.

Although Vietnamese is an alphabetic script, unlike occidental languages, as aforesaid, it is an inflectionless language in which its word forms never change, regardless of grammatical categories. This leads to a special linguistic phenomenon common in Vietnamese, called "type mutation", where a given word-form is used in a capacity that is not its typical one (e.g. a verb used as a noun, a noun as an adjective) without any morphological change (Le et al., 2010). This property introduces a huge ambiguity in Part-Of-Speech (POS) tagging, hence in automatically identifying function or content words. As a result, although function words in occidental languages are good candidates to predict boundaries of prosodic phrasing, they may not be effectively used in automatic TTS.

Besides, punctuations cannot be used as the only clue for pauses or breaks when reading a Vietnamese text. Both syllables and words in Vietnamese are separated by spaces, hence it is not easy to determine the word boundaries. In other words, Vietnamese text is a sequence of syllables, separated by spaces. This leads to a big issue in prosodic phrasing in Vietnamese TTS, which may need higher-level information from text – **syntax**. This approach was leveraged for automatic TTS by the fact that much effort had been devoted to Vietnamese syntactic parsing with good results (Le et al., 2012)(Le et al., 2009)(Nguyen et al., 2013a). The summary of syntax theory, syntactic parsing as well as the adopted Vietnamese syntactic parser, VTParser, are presented in Section 4.3. A detail of those is described in Appendix A.

In this chapter, we present proposals on prosodic phrasing using syntactic information for Vietnamese TTS. Some investigations using durational clues alone were performed to find out rules to predict pause appearance–one of the most prominent and frequent levels of prosodic phrasing, as well as lower levels of phrasing.

The analysis corpus and metrics for performance evaluation were described in Section 4.2. Section 4.4 gives an overview of our preliminary study on syntactic rules and break levels using manual syntactic parsing and a small corpus. The ultimate model, which was used in the final version of our TTS system, is covered in next sections. Section 4.5 provides general ideas to use syntactic blocks, i.e. syntactic phrases with bounded number of syllables, for predicting not only pause appearance but also final lengthening since it is a crucial aspect of the naturalness of areas around boundaries of speech (Campbell, 1993, 1992). Section 4.6 describes an improved model by grouping single-syllable blocks for final lengthening. Other features, such as syntactic links and POS tags were also used for an improved model, presented in Section 4.7 for pause appearance. The pause appearance was trained and classified using the decision tree J48 of the WEKA tool [1] for an optimized and automatic model of pause prediction.

An alpha level of 0.05 was adopted for statistical analyses.

1. a collection of machine learning algorithms for data mining tasks: http://www.cs.waikato.ac.nz/ml/weka/

4.2 Analysis corpora and Performance evaluation

4.2.1 Analysis corpora

We did a preliminary analysis on the existing corpus VNSP-Broadcaster (cf. Section 3.7 in Chapter 3). For the final proposal of prosodic phrasing modeling, the new-recorded corpora (including VDTS, VNSP-ThuTrang, and some special design utterances – called VDTO for short) were used for the analysis. For both investigations, audio files were finally time-aligned at the syllable level, and annotated for perceived pauses. Text files in our corpora were parsed to syntax trees, as in Section 4.3. They were then converted to the XML (eXtensible Markup Language) format for ease of use.

There were two main differences between the two analysis corpora: (i) the VNSP-Broadcaster corpus (existing, small and manually annotated) and (ii) the VDTO corpus (newly designed, huge, and automatically annotated).

VNSP-Broadcaster corpus. As aforesaid, the VNSP corpus included 630 utterances in about 37 minutes. Audio files were manually time-aligned at the syllable level, and annotated for perceived pauses. Text files were automatically parsed to syntax trees with constituent syntactic parsing with grammar function labels (cf. Section 4.3). These syntax trees were then manually corrected for further analysis.

Figure 4.1 illustrates an example of a syntax tree using constituent parsing with grammar function labels for the sentence "Lão muốn gì lão làm cho bằng được" *(He wanted something, he work for it at all costs)*. A hierarchical tree is shown in (a) while XML format is presented in (b). In this figure, there are grammar-functional labels, i.e. "SUB", "H", "PRD", in some phrase nodes or word leaves. These grammar-functional labels do not appear in the standard constituent parsing. If this sentence is parsed by the unnamed constituent parsing, all phrase nodes ("S", "NP", "VP") have the same name "XP", and there also are no grammar-functional labels for nodes.

VDTO-Analysis and VDTO-Testing corpora. The VDTO corpus included 5,338 utterances in about 7.7 hours. Audio files in this corpus were automatically segmented at phoneme-level by EHMM labeler. Phonemes were then grouped to syllables and perceived pauses in a different tier. Text files were transformed to syntax trees using three types of syntactic parsing by VTParser, as presented in Section 4.3.

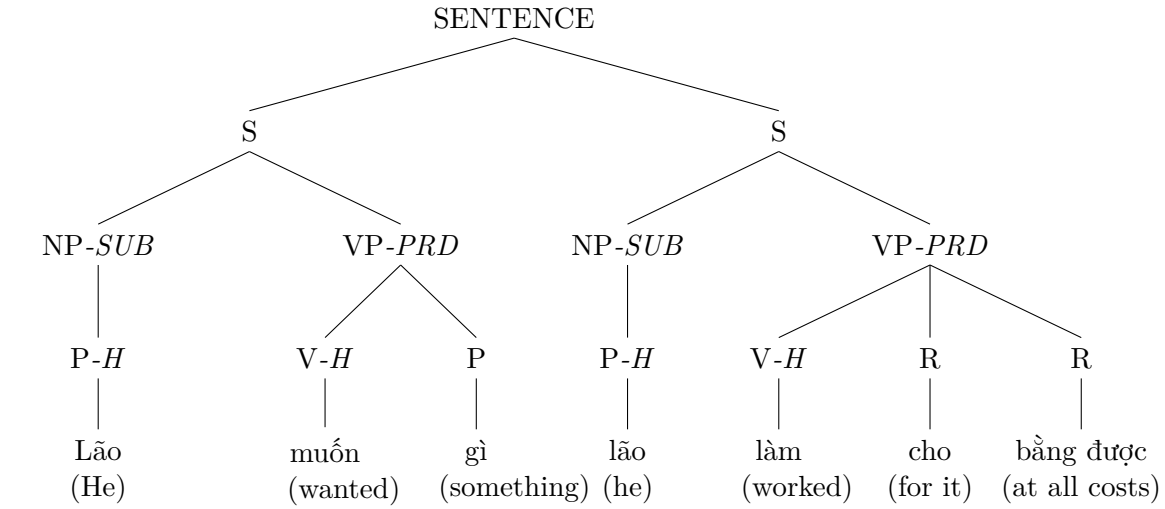
For the evaluation phase, we extracted randomly 10% of sentences in VDTO as a test corpus, called VDTO-Testing, the rest was the analysis corpus, called VDTO-Analysis. Detailed acoustic information of these corpora is summarized in Table 4.1. The analysis corpus contained about nearly seven hours of speech while the testing one had nearly one hour. Since the randomness was performed in text files, different topics and sources of VDTO were well covered in the VDTO-Testing. The mean length (syllables/sentence) of the testing corpus was a bit larger than that of the analysis one.

Syllable and pause duration.

As aforementioned, our analysis and proposal on prosodic phrasing for the Vietnamese TTS used durational clues alone. As a result, syllable and pause duration in corpora were measured. Perceived pauses were extracted and measured. In our corpora, most of them were pauses with respiratory effects. Pause durations were computed in a logarithmic scale, which was more relevant to perception.

Durations of syllables were calculated using Z-score normalization, based on the syllable

(a)



(b)

```
<SENTENCE>
   <S>
      <NP syntax="SUB">
         <N syntax="H">Lão (He)</N>
      </NP>
      <VP syntax="PRD">
         <V syntax="H">muốn (wants)</V>
         <P>gì (something)</P>
      </VP>
   </S>
   <S>
      <NP syntax="SUB">
         <N syntax="H">lão (He)</N>
      </NP>
      <VP syntax="PRD">
         <V syntax="H">làm (works)</V>
         <R>cho (for it)</R>
         <R>bằng được (at all costs)</R>
      </VP>
   </S>
</SENTENCE>
```

Figure 4.1 – An example of syntax tree using constituent parsing with grammar-functional labels: (a) hierarchical tree and (b) XML format.

Table 4.1 – VDTO analysis and test corpus

Analysis factor	VDTO-Analysis	VDTO-Testing
Number of sentences	4,805	533
Number of segments	238,584	29,149
Number of syllables	89,717	10,936
Number of pauses	9,005	1,096
Mean length (syllables/sentence)	18.67	20.52
Total duration (hours)	6.87	0.83
Total duration without pauses (hours)	6.16	0.75

structure, e.g. C1V, C1wV,C1wVC2; and tone type, i.e. long, short of the syllable, as in Formula 4.1. We adopted a hierarchical structure for Vietnamese syllables, based on an initial consonant (C1) and a rhyme. In Vietnamese, the lexical tone is carried by the rhyme on 3 elements: medial (w), nucleus (V) and ending (C2). Nucleus and tone are compulsory while others are optional. Vietnamese has a six-tone paradigm (level 1, falling 2, broken 3, curve 4, rising 5a, and drop 6a) for sonorant-final syllables, and a two-tone paradigm (rising 5b, drop 6b) for obstruent-final ones. For duration of bearing syllables, there are 2 kinds of tones: (i) long tones: 1-4, 5a, and (ii) short tones: 5b, 6a, 6b. More information can be found in Section 2.2, Chapter 2.

$$ZScore(S_i) = \frac{Length(S_i) - Mean(C_i)}{Std(C_i)} \tag{4.1}$$

where

- Length(S_i): duration length of the last syllable S_i
- C_i: Category (syllable typeand tone type) of the syllable S_i
- $Std(C_i)$: Standard deviation of the duration length of C_i

$$Std(C_i) = \sqrt{\frac{\sum_{i=1}^{N}(S_{i,j} - \overline{S_i})^2)}{N}} \tag{4.2}$$

where

- $S_{i,j}$: Duration length of the syllable j in N syllables of category C_i
- $\overline{S_i}$: Duration mean of N syllables

4.2.2 Precision, Recall and F-score

To evaluate the quality of a system or a model, several measures were adopted in this work: Precision, *Recall* and *Fscore* whose definitions can be found in some references in natural language processing. In this work, these measures are used to state the quality of a syntax parser or a model of pause prediction.

Recall (also called positive predictive value, illustrated in Formula 4.4) is the proportion of Real Positive (RP) cases that are correctly Predicted Positive (PP). That is the coverage of the real positive cases by the predicted positive rule. Conversely, Precision (also known as sensitivity, illustrated in Formula 4.3) computes the proportion of predicted positive cases that are correctly real positives. True and False Positives (TP/FP) refer to the number of

predicted positives that were correct/incorrect (Powers, 2007).

$$Precision = \frac{TP}{PP} = \frac{TP}{TP + FP} \tag{4.3}$$

$$Recall = \frac{TP}{RP} \tag{4.4}$$

where

- FP: Number of False Positive;
- TP: Number of True Positive
- PP: Number of Predicted Positive;
- RP: Number of Real Positive

A measure that combines precision and *Recall* is the harmonic mean of precision and *Recall*, the traditional F-measure or balanced $F - score$ (F), illustrated in Formula 4.5. This is also known as the F_1 measure, because *Recall* and precision are evenly weighted, providing a single measurement for a system. It is a special case of the general F_β measure (for non-negative real values of β) in Formula 4.6. Two other commonly used F measures are the F_2 measure, which weights *Recall* more than precision, and the $F_{0.5}$ measure, which puts more emphasis on precision than *Recall* (Clark et al., 2013).

$$F - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.5}$$

$$F_\beta = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall} \tag{4.6}$$

where

- β: $\beta > 0$ times as much importance to *Recall* as precision
  - $F_{0.5}$: weight precision twice as much as *Recall*
  - F_2: weight precision twice as much as *Recall*
  - F_1 or $Fscore$: the same weight for precision and *Recall*, for short F

4.2.3 Syntactic parsing evaluation

In syntax parsing, the evaluation technique that is currently the most widely-used was proposed by the Grammar Evaluation Interest Group (Grishman et al., 1992), and is often known as "PARSEVAL". It is basically a relaxation of full identity as the success criterion to one which measures similarity of an analysis to a test corpus analysis. The original version of the scheme utilised only phrase-structure bracketing information from the annotated corpus and compares bracketings produced by the parser with bracketings in the annotated corpus.

Due to the ease of comparison, the evaluation method from the work of Collins (1999) and Bikel (2004), is adopted in this work, which measures how much the elements (constituents or dependents) in the hypothesis parse tree look like the constituents in a hand-labeled gold reference parse. In other word, the method compares elements produced by the parser

with elements in the annotated corpus (TreeBank) and computes the number of matched element ME with respect to the number of elements PE returned by the parser (expressed as *Precision*, Formula 4.7) and with respect to the number CE in the corpus (expressed as *Recall*, Formula 4.8) per sentence.

$$Precision = \frac{ME}{PE} \tag{4.7}$$

$$Recall = \frac{ME}{CE} \tag{4.8}$$

where

- ME: Number of **M**atched **E**lements
- PE: Number of **E**lements returned by the **P**arser
- CE: Number of **E**lements in the **C**orpus

4.2.4 Pause prediction evaluation

In pause prediction, Precision (P) was the probability that a (randomly selected) predicted pause was an actual (correct) pause in corpus., i.e. the fraction of the number of correct predicted pauses to the total number of actual pauses in corpus (Formula 4.9). Recall (R), the probability that an (randomly selected) actual pause in corpus is predicted, was calculated as the number of correct predicted pauses over the total number of actual pauses in corpus (Formula 4.10) (Taylor, 2009).

$$Precision = \frac{CP}{PP} \tag{4.9}$$

$$Recall = \frac{CP}{AP} \tag{4.10}$$

where

- PP: Number of **P**redicted **P**auses
- CP: Number of **C**orrect predicted **P**auses
- AP: Number of **A**ctual **P**auses in the middle of utterances in corpus

4.3 Vietnamese syntactic parsing

This section recapitulates the syntax theory, Vietnamese grammatical categories and syntactic structure, and the current state of Vietnamese syntax parsing. The adoption of the state-of-the-art technique for Vietnamese syntactic parsing is described with a parser – VTParser for the TTS system. A detail of these is presented in Appendix A.

4.3.1 Syntax theory

Grammar, composing syntax and morphology, helps us analyze and describe the word and sentence patterns of a language by formulating a set of rules with respect to those patterns. Morphology is the study of the form and structure of a given language's morphemes and

other linguistic units. Whereas, studying syntax provides us how to construct sentences, and a number of possible arrangements of the elements in sentences (Kroeger, 2005).

Grammatical categories, a natural first step toward allowing grammars to capture word generalizations, covers not only the Part Of Speech (POS), e.g. noun, verb, preposition but also types of phrase, e.g. noun phrase, verb phrase, prepositional phrase. Parts of speech are termed as lexical categories or word classes whereas non-lexical categories or phrasal categories means types of phrase. Two major aspects of sentence syntactic structure are phrase structure grammar and dependency grammar. The first aspect concerns the organization of the units that constitute sentences, hence also referred as constituency structure grammar, e.g. Sentence → Prepositional phrase + Noun phrase + Verb phrase. The second one, dependency grammar, concerns the function of elements (i.e. dependency relations) in a sentence such as subject, predicate or object, which have traditionally been referred to as grammatical relations or relational structure.

Grammatical categories. To classify words into "*grammatical categories*" is a natural first step toward allowing grammars to capture generalizations. The term "*grammatical category*" now covers not only the Parts Of Speech (POS), e.g. nouns, verbs, prepositions but also types of phrase, e.g. noun phrases, verb phrases, prepositional phrases. Parts of speech are termed as "*lexical categories*" in contemporary linguistics or traditionally referred as "*word classes*", whereas "*non-lexical categories*" or "*phrasal categories*" means types of phrase (Valin, 2001)(Kroeger, 2005). The most important lexical categories are nouns, verbs (V), adjectives (A), adverb (R) and prepositions (E). Nouns can be categorized in numerous ways, e.g. proper nouns (Np, i.e. proper name), common nouns (N, i.e. not refer to unique individuals or entities). Pronouns (P) are closely related to nouns, and "traditionally characterized as substitutes for nouns or as standing for nouns".

Phrase structure grammar. A sentence does not consist simply of a string of words; and not the case that "each word is equally related to the words adjacent to it in the string" (Valin, 2001). Words in a sentence may be grouped into grammatical units of various sizes. One crucial unit is the clause, "the smallest grammatical unit which can express a complete proposition". A sentence may consist of just one clause or several clauses. A single clause may contain several phrases, another important unit. A single phrase may contain several words, which may contain several morphemes. "Each well-formed grammatical unit (e.g. a sentence) is made up of constituents which are themselves well-formed grammatical units", such as clauses, phrases, etc. There are only a limited number of basic types of units, which is adequate for a large number of languages: sentence, clause, phrase, word, and morpheme. This kind of structural organization is called a part–whole hierarchy: each unit is entirely composed of smaller units (Kroeger, 2005, p. 32-33).

There are two basic ways in which one clause can be embedded within another: coordination vs. subordination. In a coordinate structure, two constituents belonging to the same category are conjoined to form another one of that category. In a coordinate sentence, two (or more) main clauses (or independent clauses – S) occur as daughters and co-heads of a higher clause. A dependent clause (or subordinate clause – SBAR, i.e. complement clauses, adjunct or adverbial clauses and relative clauses) is one that functions as a dependent, rather than a co-head. This combination of words cannot stand-alone or form a complete sentence, but provides additional information to finish the thought (Kroeger, 2005).

The term "phrase" in linguistics has a more precise meaning other than "any group of words". That is a group of words that function as a constituent (i.e. a unit for purposes of

word order) within a simple clause. Phrases may be classified into different categories, such as noun phrases (NP), verb phrases (VP). There exists one word in most phrases being the most important element of the phrase, called the *head* (H) of the phrase. The category of phrase heads in general gives name to the phrase.

The following example illustrates this hierarchical structure in Vietnamese: [$_S$ [$_{NP}$ [$_N$ Cô giáo *(The teacher)*] [$_{Np}$ tiếng Anh *(English)*] [$_{SBAR}$ mà *(who)* [$_{NP}$ [$_N$ anh *(you)*]] [$_{VP}$ đã [$_V$ gặp *(met)*]] [$_{NP}$ [$_N$ hôm qua *(yesterday)*]] $_{SBAR}$] $_{NP}$] [$_{VP}$ đang [$_V$ đọc] *(is reading)* [$_{NP}$ [$_N$ sách *(books)*]] [$_{PP}$ [$_P$ trong *(in)*] [$_{NP}$ [$_N$ thư viện *(the library)*]$_{NP}$]$_{PP}$] $_{VP}$] $_S$].

Dependency structure grammar.

Another important aspect of sentence structure need to be considered, namely "*grammatical relations*". Those are the *syntactic function* of elements such as subjects or objects in a sentence. Therefore, this type of syntax is referred to as "*relational structure*". This is also termed as "*dependency structure*" since it actually encompasses the dependency relation.

Aside from the predicate itself, the elements of a simple clause, i.e. clausal dependents, can be classified as either adjuncts or arguments, illustrated in Figure A.2. Adjuncts (ADT) are elements that are "not closely related to the meaning of the predicate but which are important to help the hearer understand the flow of the story, the time or place of an event, the way in which an action was done, etc". Adjuncts can be omitted without creating any sense of incompleteness. Arguments are those elements that are "selected by the verb"; they are "required or permitted by certain predicates, but not by others". In order to be expressed grammatically, arguments must be assigned a grammatical relation within the clause. There are two basic classes of grammatical relations: obliques (or indirect arguments) vs. terms (or direct arguments). Terms (i.e. subject–SUB, primary object–OBJ, secondary object–OBJ2) "play an active role in a wide variety of syntactic constructions", while obliques (OBL) are "relatively inert" (Kroeger, 2005, p. 62).

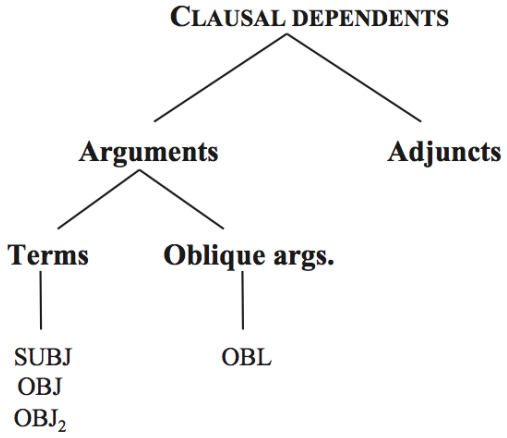


Figure 4.2 – Classification of clausal elements (Kroeger, 2005, p. 62).

Some clausal dependents are illustrated in the following example for Vietnamese: [$_S$[$_{ADT}$ Tối qua *(last night)*] [$_{SUB}$ Kiên *(Kien)*] [$_{PRD}$ đã tặng *(gave)* [$_{OBJ}$ một bó hoa hồng *(a bouquet of roses)*] [$_{OBL}$ cho mẹ của anh ấy *(to his mother)*] $_{PRD}$] $_S$].

4.3.2 Vietnamese syntax

In order to address problems of syntactic parsing (i.e. syntactic analysis, cf. Section A.2, Appendix A), a common way is to construct a *treebank*. A treebank is simply a collection of sentences (normally a large sample of sentences, also called a corpus of text), where each sentence is provided by a complete syntactic analysis.

Treebank solves the knowledge acquisition problem (i.e. designing out a grammar to cover all syntactic analysis of natural language) by finding the grammar underlying the syntax analysis. Obviously, there is no set of syntactic rules or linguistic grammar, as well as there is no list of syntactic constructions provided explicitly in a treebank. In fact, the parser can infer a set of implicit grammar rules to cover a large amount of syntactic analysis that does not exist in treebank. Concerning the problem of explosion of rule combinations, since each sentence in a treebank has been given its most plausible syntactic analysis, some supervised learning methods can be used to train a scoring function over all possible syntactic analyses of that sentence. For a given sentence that is not seen in the training data, a statistical parser can use this scoring function to return the syntax analysis that has the highest score, which is taken to be the most plausible analysis for that sentence.

The syntactic parsing for each sentence should have been annotated by human expert to guarantee the most plausible analysis for that sentence. Before the annotation process, an *annotation guideline* is typically written in order to ensure a consistent scheme of annotation throughout the treebank.

This section presents VietTreebank, a Vietnamese TreeBank (Nguyen et al., 2009), and the Vietnamese syntax that the VietTreebank used and followed for the annotation.

Vietnamese TreeBank.

Vietnamese treebank (VietTreebank) (Nguyen et al., 2009) was constructed as a result of a national project in Vietnam, VLSP (Vietnamese Language and Speech Processing) [2]. The construction of this corpus included five major phases: (i) investigation, (ii) guideline preparation, (iii) tool building, (iv) raw text collection, and (v) annotation. Raw texts were collected from the Youth online daily newspaper, with a number of topics including social and politics. To the best of our knowledge, despite various existing issues, up till now, VietTreebank has been the only corpus used in natural language processing for Vietnamese.

Table 4.2 – VietTreebank corpus (Nguyen et al., 2009, p. 14)

Data set	Sentences #	Words #	Syllables #
POS tagged	10,368	210,393	255,237
Syntactically labeled	9,633	208,406	251,696

POS tag set. Since Vietnamese word order is quite fixed, a phrase structure representation was chosen for syntactic structures in VietTreebank. There were three annotation levels including word segmentation, POS tagging, and syntactic labeling. The word segmentation identified word boundary in sentences. The POS tagging assigned correct POS tags to words. The syntactic labeling recognized both phrase-structure tags and functional tags. Table 4.2 shows the sizes of the two data sets in this corpus: (i) The data set tagged with POSs: 10,368 sentences, and (ii) The data set annotated with syntactic labels: 9,633 sentences.

In this work, we adopted the Vietnamese POS tag set from the work of Le et al. (2010),

2. http://vlsp.vietlp.org:8080/

illustrated in Table 4.3. This complete tag set was designed to use for annotating the Vietnamese treebank (Nguyen et al., 2009).

Table 4.3 – Vietnamese POS tag set (Le et al., 2010, p. 14)

No.	Category	Description	No.	Category	Description
1.	Np	Proper noun	10.	M	Numeral
2.	Nc	Classifier	11.	E	Preposition
3.	N	Common noun	12.	C	Subordinating conjunction
4.	P	Pronoun	13.	CC	Coordinating conjunction
5.	Nu	Unit noun	14.	I	Interjection
6.	V	Verb	15.	T	Auxiliary, modal words
7.	A	Adjective	16.	Y	Abbreviation
8.	R	Adverb	17.	Z	Bound morpheme
9.	L	Determiner	18.	X	Unknown

Major lexical categories in Vietnamese are noun (including common noun N, classifier Nc, proper noun Np, unit noun Nu, pronoun P), verb (V), adjective (A), adverb (R) and preposition (E). Minor ones are conjunction (subordinating C, coordinating CC), determiner (L), numeral (M), interjection (I), auxiliary/modal words (T), and bound morpheme (Z). Proper nouns (Np) can be Vietnamese proper names, e.g. "Hà Nội", "Nguyễn Khuyến", or loanwords, e.g. "Luân-Đôn" (London), "Ê-li-da-bét" (Elizabeth). Examples for common nouns (N), i.e. not refer to unique individuals or entities, are "bàn" (table), "mèo" (cat), "ghế" (chair), etc.

Beyond the classical POS used in Western languages (noun, verb,...), there does exist the presence of classifiers, which are commonly found in Asian languages. Classifiers are independent words considered as nouns, which "occupy a special position in the noun phrase, but do not seem to contribute to the meaning of the noun phrase in any definite way" (Kroeger, 2005). The classifier may possibly categorize referents (normally nouns) based on their attribute such as shape, function, or animacy. Unlike European languages, in general, Vietnamese common nouns are required to be accompanied by a classifier, and vice versa since the meaning of a Vietnamese classifier cannot be specified in isolation. Vietnamese is one of several Asian languages with a complex numeral classifier system. In English, most nouns need to be chosen between a singular and a plural (e.g. table vs. tables) whereas Vietnamese nouns "do not in themselves contain any notion of number or amount. In this respect they are all somewhat like English mass nouns such as milk, water, flour, etc." (Thompson, 1987, p. 193). Vietnamese classifiers can be used in "anaphoric construction where classifiers are considered as a pronoun to replace the omitted head noun" (means "one"), such as "cái lớn" (a big one). Two most commonly used classifiers in Vietnamese language rare "con" (for animate, non-human objects) and "cái" (for inanimate objects) (Dao, 2011). Major Vietnamese classifiers are presented in Appendix A.

Nouns can also be accompanied by a determiner (L), such as "mấy cái chìa khoá" *(some keys)*, "nhiều cửa sổ" *(many windows)*, "những ngôi nhà" *(houses)*, "chút tiền" *(a little money)*; or a numeral (M) such as "ba chiếc kẹo" *(three candies)*. Pronouns in Vietnamese may substitutes for nouns, such as "đó", "đấy", "ấy", "kia" *(that)*, "đây", "này" *(this)*. First- and second-person pronouns are much more complicated than other European languages, since they depends on the relationship, gender or ages of speakers and listeners. For instance, the pronoun pair of "I-you" in English can be "tôi-bạn" between two persons with a general relationship (i.e. ignoring gender, ages...), "mẹ-con" between mothers and childs, "ông-cháu"

between grandfathers and grandchilds, "mày-tao" between two persons with a close or negative relationship. The genders and ages also plays an important role to decide pronouns for the second-persons, such as "chị" *(sister)* for older females, "cô" *aunt* for much older females and "bà" *grandmother* for much much older females; while "anh" *(brother)*, "chú" *(uncle)* and "ông" *(grandfather)* for males respectively. The last sub-type of nouns is unit noun, which shows a unit or a measure, such as "phút" *(minute)*, "mét" *(meter)*, "km/h", ect.

'Bound morphemes" (Z) designate syllables that are not supposed to appear alone and should only be seen as part of a compound word, and this tag is normally only ever used to deal with cases when the segmentation of the corpus has been done improperly. Some examples of adjectives in Vietnamese include "to" *(big)*, "dài" *(long)* or "mỏng" *(thin)* for sizes; "tròn" *(circle)* or "vuông" *(square)* for shapes; "đắng" *(bitter)*, "tươi" *(fresh)* or "cay" *(spicy)* for tastes; "xấu" *(ugly)*, "mềm" *(soft)* or "chính xác" *(correct)* for qualities. Some common Vietnamese adverbs are "vẫn" *(still)*, "chưa", "không" *(not)*, "quá" *(too)*, "rất" *(very)*, "thật" *(really)*.

Syntactic structure.

Two types of syntactic structure, i.e. constituency and dependency structure grammar, were annotated in VietTreebank. However, the constituency representation, i.e. phrase structure, was chosen as the main structure using brackets since Vietnamese has a quite fix word order. Dependency relations were annotated by functional labels for corresponding constituents. Independent clauses, i.e. main clauses, were labeled as "S" whereas "SBAR" was annotated for dependent clauses, i.e. subordinate clause (including complement clauses, adjunct or adverbial clauses and relative clauses).

A phrase includes one or more heads (phrase head–H, a functional label generally giving name to the phrase), preceding and succeeding supplement elements. For instance, the common noun "người" *(person)*, which determines the phrase name (noun phrase - NP), is the phrase head of "một **người** cao lớn" *(a tall and big **person**)*. The preceding supplement element, "một" *(a)*, is a numeral (M) while the succeeding one, "cao lớn" *(tall and big)*, is an adjective (A). Phrases whose head words are common nouns, classifiers, proper nouns, unit nouns, or pronouns are noun phrases. Some other main phrasal categories are PP (prepositional phrase), VP (verb phrase), AP (adjective phrase) and RP (adverb phrase). In addition, QP was also adopted for numeral phrases; UCP refers a phrase including two or more head elements in different categories, connected by a coordinating conjunction (CC). Other phrases were labeled as XP, such as expressions or other unclassified phrases. Some examples of Vietnamese phrases are shown below.

Main functional labels. i.e. dependency relations, in VietTreebank are "SUB" for subjects, "PRD" for full predicates, "H" for phrase heads, "DOB" for direct objects, "IOB" for obliques. Adjuncts are annotated by a list of labels, which shows their semantic functions that is "TMP" for time (temporal adjunct), "LOC" for location (locative adjunct), "MNR" for manner (modificative adjunct), "CND" for condition (conditional adjunct), "PRP" for purpose (causal adjunct), etc. Other semantic functions of adjuncts are annotated as "ADV".

Phrase structure in VietTreebank is represented by brackets, which are straightforwardly converted to hierarchical trees. Functional labels are labeled as properties of constituent elements' nodes. Figure 4.3 illustrates an example of the sentence "Men theo con đường mòn, chúng tôi đến một khu đất trước dãy núi Sen" *(Skirting a rut, we went to a piece of land before a row of the mountain Sen)* using (a) brackets (b) a hierarchical tree.

(a)

```
(S
    (VP-ADV (V-H Men)
            (PP-MNR (E theo)
                    (NP (Nc-H con) (N đường mòn))
            )
    )
    (, ,)
    (NP-SUB (P-H chúng tôi))
    (VP (V-H đến)
            (NP-DOB (M một)
                    (N-H khu)
                    (N đất)
                    (PP-LOC (E-H trước)
                            (NP (N-H dãy)(N núi)(Np Sen))
                    )
             )
    )
    (. .)
)
```

(b)

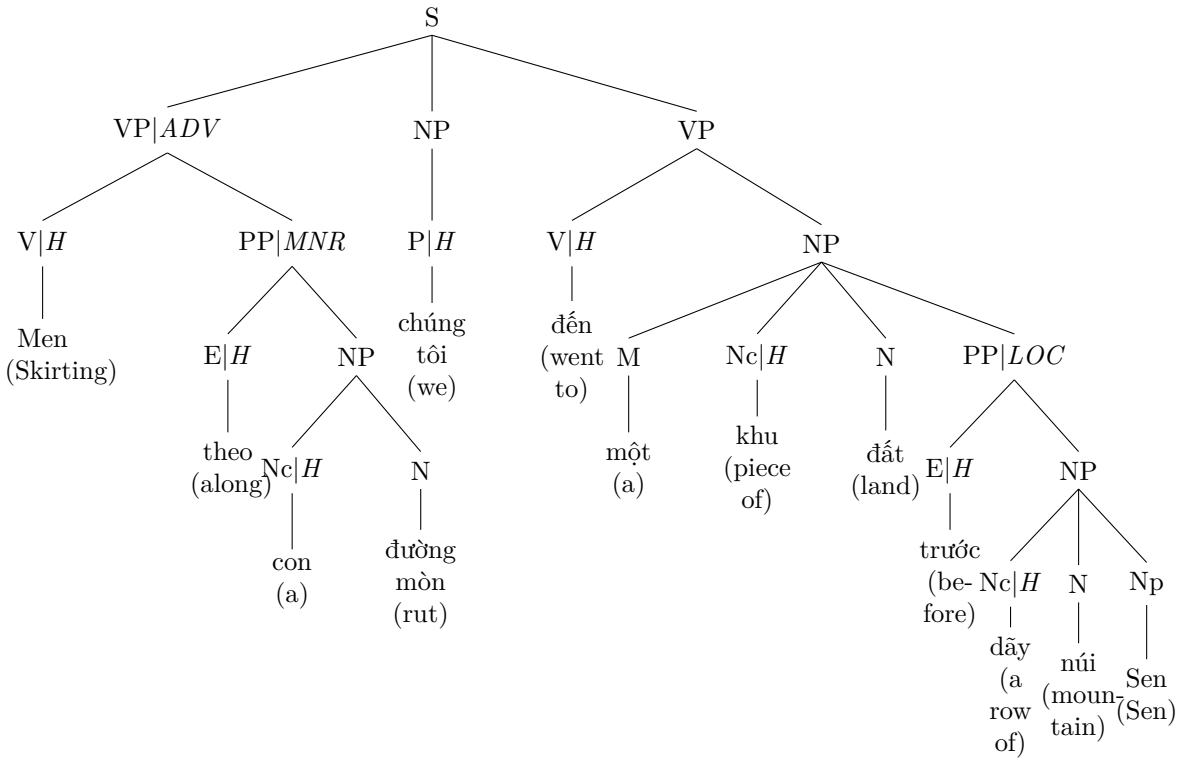


Figure 4.3 – An example of a sentence annotated in VietTreebank using: (a) brackets and (b) a hierarchical tree.

Example 7 Some examples of Vietnamese phrases

- NP: những [$_{Nc}$ quả] bóng màu xanh *(green [classifier] balls)*
- PP: "[$_{E}$ trên] mặt đất" *(on the ground)*, "[$_{v}$čE từ] năm 1990" *(since 1990)*, "[$_{E}$ của] tổ quốc ta" *(of our fatherland)*
- VP: "hay [$_{V}$ đi chơi] với bạn bè" *(often go out with friends)*, "[$_{V}$ bắt đầu] làm việc từ sớm" *(start working early)*
- AP: "rất [$_{A}$ đẹp]" *(very beautiful)*, "[$_{A}$ giỏi] về thể thao" *(good at sport)*
- QP: "hơn [$_{M}$ 200]" *(more than 200)*
- RP: "[$_{R}$ vẫn] chưa" *(still not)*
- UCP: "vải [$_{UCP}$ [$_{AP}$ **rẻ**] và [$_{NP}$ chất tốt]]" *(**cheap** and good quality clothes)*
- XP: "ba cọc ba đồng" *(fixed and modest–for income)*

4.3.3 Syntactic parsing techniques

In natural language processing, the syntactic analysis (hereafter called syntactic parsing) may vary from low to high levels. The lowest level can be referred as simply part-of-speech tagging for each word in the sentence. Shallow parsing (also known as "chunking", "light parsing") decomposes of sentence structure into constituents but not specify their internal structure nor their role in the main sentence. The highest level parsing, i.e. the full parsing, can recover not only the phrase structure of a sentence, but also can identify the sentence structure dependency between each predicate in the sentence and its explicit and implicit arguments. In syntactic parsing, ambiguity is a particularly onerous issue since the most probable analysis has to be chosen from an exponentially large number of alternative analyses. As a result, parsing algorithms plays an important role to handle such ambiguity, hence decides the quality of a parser corresponding to different levels from tagging to full parsing.

A detail of main syntactic parsing techniques is presented in Appendix A. **Generative models**. The main idea of generative models is that in order to find the most plausible parse tree, the parser has to choose between the possible derivations, each of which can be represented as a sequence of decisions. Probabilistic Context-free Grammars (PCFG) model is the simplest classical instance of generative models, where the parse tree has the highest joint probability with the input sentence. The most popular and classical generative model is Lexical Probabilistic Context-free Grammars (LPCFG) of Collins (1999). Its idea is to extend the history of a parse tree by adding more information of phrase head words. On the test set, that is the section 23 of English Penn Treebank (Marcus et al., 1993), the LPCFG parser can reach $Fscore$ of 88.2%, while $Fscore$ of the parser with the naive PCFG is 73%.

The current state-of-the-art generative parsing model in terms of accuracy belongs to the well-known Berkeley parser (Petrov and Klein, 2007). These authors assumed that if it is possible to split each constituent label (even POS) in Treebank in a good manner, a high accuracy can be obtained. This method is called *"Latent Variables PCFG"*, which uses the Expectation-Maximization (EM) algorithm to find the best manner to split their grammar,

reaching $Fscore$ of 90.1%. In syntactic parsing, Berkeley parser has been considered as one of the strongest one because it does not need any grammar information, only the Treebank corpus, making it easily apply into any languages.

Discriminative models. The definition of PCFG means that various rule probabilities had to be adjusted in order to obtain the right scoring of parses. Meanwhile, the independence assumptions in PCFG, which are dictated by the underlying CFG, often leads to bad models. These models cannot use information vital to the decision of rule scores leading to high scoring plausible parses. Such ambiguities can be modeled using arbitrary "features" of the parse tree. Discriminative methods provide us with such a class of models. Even in common machine learning, the performance of the discriminative models is usually better than that of the generative models.

Collins (2002) created a simple framework that described various discriminative approaches to train a parsing system (and also chunking or tagging). This framework was called a *global linear model* (Collins, 2002). Commonly, a *conditional random field* (Lafferty et al., 2001) could be used to define the conditional probability as a linear score for each candidate and a *global* normalization term. However, a simpler global linear model can be obtained by ignoring the normalization term (thus much faster to train). Many experimental results in parsing have shown that this simpler model often provides the same or even better accuracy than the more expensively trained normalized models.

Advanced parsing methods

Beside the above learning models, there are a number of advanced methods that utilized the external information to boost the performance of parsing systems to higher levels. Socher et al. (2013) used the deep learning technique, which was based on the recurrent neural network, and reach the $Fscore$ of 90.5%. Charniak and Johnson (2005) proposed a general framework called Re-ranking parser. This framework first used a baseline generative parser (such as one in Collins (1999) or in Petrov and Klein (2007)) to produce top k-best candidate parse trees, and then used a discriminative model with a set of strong and rich features to re-rank them and pick out the best one. This work used maximum entropy model as a discriminative re-ranker for the baseline system, which could achieve a high $Fscore$ of 91.5% on test set of English Treebank. Huang (2008) improved the strategy for the re-ranking parsers that could encode more candidate parse trees in the first phase and utilize the averaged perceptron model to perform the re-ranking phase, reaching up to $Fscore$ of 91.8% on English test set.

However that is not whole story, McClosky et al. (2006) even extended the idea of re-ranking parser by injecting more unsupervised features from large external text corpus, making the parser become a self-trained system that could achieve a $Fscore$ of 92.4% on the test set. Currently, the self-trained parser has been considered as the state-of-the-art parsers in terms of $Fscore$ on the English test set.

4.3.4 Adoption of parsing model

Averaged perceptron. A well-known discriminative model, **Perceptron**, was adopted for the Vietnamese syntactic parsing in our TTS system. A perceptron (Rosenblatt, 1988) originally introduced as a single-layered neural network. In structured prediction problem such as parsing, perceptron could be considered as the most widely-used model due to its simplicity and efficiency. Comparing to the generative model or other discriminative models, it is much simpler while still keeping a competitive accuracy (Carreras et al., 2008, Collins and Roark,

2004, Zhu et al., 2013). Perceptron could be trained by using the online learning, that is, processing examples one at a time, during which it adjusts a weight parameter vector that can then be applied on input data to produce the corresponding output. The weight adjustment process awards features appearing in the truth and penalizes features not contained in the truth. After the update, the perceptron ensures that the current weight parameter vector is able to correctly classify the present training example.

A detail learning algorithm of the original perceptron, voted perceptron, and averaged perceptron are described in Appendix A. Although the **original perceptron** learning algorithm is simple to understand and to analyze, the incremental weight updating suffers from over-fitting, which tends to classify the training data better, at the cost of classifying the unseen data worse. Also, the algorithm is not capable of dealing with training data that is linearly inseparable. Freund and Schapire (1999) proposed a variant of the perceptron learning approach, called the **voted perceptron** algorithm. Instead of storing and updating parameter values inside one weight vector, its learning process keeps track of all intermediate weight vectors, and these intermediate vectors are used in the classification phase to vote for the answer. The intuition is that good prediction vectors tend to survive for a long time and thus have larger weight in the vote. Compared with the original perceptron, the voted perceptron is more stable, due to maintaining the list of intermediate weight vector for voting. Nevertheless, to store those weight vectors is space inefficient. Also, the weight calculation, using all intermediate weight parameter vectors during the prediction phase, is time consuming. The **averaged perceptron** algorithm (Freund and Schapire, 1999) is an approximation to the voted perceptron that, on the other hand, maintains the stability of the voted perceptron algorithm, but significantly reduces space and time complexities.

It turns out that the perceptron, especially averaged one, is one of the most powerful model in parsing and in resolving different problems in natural language processing. Collins and Roark (2004) reported that a incremental parsing with the use of averaged perceptron could reach a comparable $Fscore$ (86.6%) comparing to the generative model (86.7%). Zhu et al. (2013) shew that perceptron-based parser could achieve $Fscore$ of 90.4%, which outperformed the currrent state-of-the-art generative parser (Petrov and Klein, 2007) without using any latent variables. Carreras et al. (2008) proposed a way of using Tree Adjoining Grammar (TAG) with the use of perceptron algorithm, which could produce a parsing accuracy of 91.1%, certainly one of the state-of-the-art accuracy in parsing technique.

Shift-reduce parsing with averaged perceptron. We adopted the parsing model and a syntactic parser of Le et al. (2015) on constituency parsing for our TTS system. Similar to any practical problem, there are two criteria for a parser in speech synthesis: the accuracy and the parsing speed. Therefore, it is necessary to select a system that can balance both of them. Some experiments were done in the work of Le et al. (2015) to compare the state-of-the-art parsers in terms of their performance on the test set of English Treebank (Marcus et al., 1993).

Table 4.4 presents the result of such experiments, including both their averaged speed and F-score. The result shows that the generative parsers had lower performances than the discriminative ones. Only Petrov and Klein (2007) could achieve a high accuracy, but their speed was quite slow (6.1 sentence/s), especially when applying into practical problems such as speech synthesis. The most accurate parsers belonged to a group using the advance models. However, due to the complexity of these models, their speeds were also very slow and they

Table 4.4 – F-score of the adopted parsing system on English Test Set comparing with state-of-the-art parsers

Model	System	Speed (sentence/s)	LR	LP	F1
Advanced models	Charniak and Johnson (2005)	2.1	91.2	91.8	91.5
	McClosky et al. (2006)	1.2	92.2	92.6	92.4
	Socher et al. (2013)	3.3	90.3	90.7	90.5
	Huang (2008)	N/A	92.2	91.2	91.7
Generative models	Petrov and Klein (2007)	6.1	90.1	90.3	90.2
	Collins (1999)	3.5	88.1	88.3	88.2
	Sagae and Lavie (2005)	3.7	86.0	86.1	86.0
	Sagae and Lavie (2006)	2.2	88.1	87.8	87.9
Discriminative models	Carreras et al. (2008)	N/A	90.7	91.4	91.2
	Zhu et al. (2013)	32.4	90.2	90.7	90.4
	Zhu et al. (2013) + semi	11.6	91.1	91.5	91.3
	Charniak (2000)	5.7	89.5	89.9	89.5
	Adopted parsing model	**13.6**	**90.9**	**91.2**	**91.1**

even required external resources outside of the scope of Treebank, which was expensive to prepare.

In the work of Le et al. (2015), the parsing system of Zhu et al. (2013) was selected as a baseline system to extend due to the following reasons. First, that system could achieve a state-of-the-art accuracy with a fast parsing speed. Second, the parser had been trained using an averaged perceptron, a global linear model, which was a simple yet fast training model and could be easily to perform an online training. In addition, the baseline system was based on the shift-reduce parsing algorithm Zhang and Clark (2009), which could perform in linear time complexity with richer feature set than the traditional chart-based parsing algorithm (Carreras et al., 2008, Charniak, 2000, Collins and Roark, 2004). As a result, this system could achieve a good balance between the parsing speed and the F-score accuracy. However, it relied on an inexact search such as beam-based method (Zhu et al., 2013) in both training and parsing phase. This bottleneck might lead to a search error, causing some lost on accuracy.

In order to reduce the search error, Le et al. (2015), our adopted parsing model, proposed a method in which an exact search was performed instead of the approximation method. The main idea in this work was to use dynamic programming (Huang and Sagae, 2010) and A* search (Klein and Manning, 2003) to guarantee the optimality. The system of Le et al. (2015) was the first parsing system that could perform an exact search for shift-reduce parsing with the averaged perceptron algorithm. As shown in Table 4.4, the fastest model with 32.4 sentences/s has F-score of 90.2%. The adoption system could achieve a high F-score of 91.1% with a high parsing speed (13.6 sentences/s). Zhu et al. (2013) can achieve a little bit better accuracy (91.3%) than the adopted system but more external semi-supervised features had to be used to fulfill the blank of search errors (hence lower speed at 11.6 sentence/s).

4.3.5 VTParser, a Vietnamese syntactic parser for TTS

Despite a considerable gap in quality to other common languages such as Chinese, Japanese or English (e.g. F-score for English = 90%-92%); there have been at least three most popular

syntactic parsers for Vietnamese: vnLTAGParser (Le et al., 2012), LPCFG parser (Le et al., 2009) and Berkeley parser for Vietnamese (Nguyen et al., 2013a). The vnLTAGParser, adopting the Lexicalized Tree-Adjoining Grammars for both constituency and dependency parsing, had the F-score of 73.21% (dependency) and 69.33% (constituency). The best F-score in Le et al. (2009) was around 78% (constituency). The last system, which was originally the Berkeley parser but applied for Vietnamese language (Nguyen et al., 2013a), provided a F-score of 73.21% (dependency).

There were two major reasons for such "less-than-stellar" performance. First, Vietnamese Treebank corpus was not sufficient, stable and accurate enough to be able to produce a good parsing model. Second, most of the existing Vietnamese were relied much on the generative parsing models, which has been empirically proven to be less accurate than most of the state-of-the-art parsing models.

VTParser, a Vietnamese syntactic parser for TTS, was built based on the parsing model of Le et al. (2015) for the constituency parsing, trained with the VietTreebank corpus, with some adaptions for the Vietnamese language. An experiment had been performed to evaluate the effectiveness of VTParser in the Vietnamese parsing, compared to three above well-known Vietnamese parsers including: (i) the Vietnamese Berkeley parser (Petrov and Klein, 2007), (ii) the LPCFG parser (Le et al., 2009), and (iii) the vnLTAGParser (Le et al., 2012). These four syntactic parsers in this experiment were trained using the VietTreebank corpus with the same train-test split as the work of Le et al. (2009).

Table 4.5 – Results of experiment comparing between different Vietnamese parsers

System	**Speed** (sentences/s)	**F-score**
LPCFG parser	4.3	78.2%
Berkeley parser	6.2	71.5%
vnLTAGParser	N/A	69.33%
VTParser	**13.6**	**81.6%**

Table 4.5 presents the experimental result, showing that the VTParser had outperformed all other Vietnamese parsers in both parsing accuracy and speed. This parser was about twice quicker than the quickest one (vnLTAGParser), and about 3.4% higher than the one having the best accuracy (LPCFG parser).

Based on the different approaches of prosodic phrasing modeling, presented in Chapter 4, we proposed three types of syntax parsing for our TTS system as follows.

- **Standard constituency parsing:** Sentences are parsed into syntax trees. Leaves of these trees are grammatical words named by POS categories while ancestor nodes are syntactic phrases named by phrasal categories. This is a standard parsing of phrase structure.

- **Unnamed constituency parsing:** Sentences are parsed into syntax trees. Leaves of these trees are grammatical words named by POS categories while ancestor nodes are unnamed syntactic phrases. This type of syntactic parsing was proposed especially for our work on prosodic phrasing modeling using phrase structure but not phrase names (cf. syntactic blocks in Section 4.5).

- **Constituency parsing with grammar-functional labels:** Sentences are parsed into syntactic trees as the standard constituent parsing. However, there is additional infor-

Table 4.6 – Experimental results of three syntax parsing types for Vietnamese

Syntax parsing type	P	R	F-score
Standard constituent parsing	81.14%	82.72%	81.61%
Unnamed constituent parsing	84.43%	85.40%	84.61%
Constituent parsing with functional labels	70.56%	72.33%	71.11%

mation for some nodes in syntax trees. Some phrase nodes or word leaves are assigned with some special necessary grammar-functional labels: main clause (S), subordinate clause (SB), adjuncts (ADT) for all semantic functions, head of phrase (H), subject (SUB) and predicate (PRD). The averaged perceptron algorithm was also used to train using VietTreebank for the grammar-functional labeling. This type of parsing was used for our preliminary analysis on syntactic rules and break levels (cf. Section 4.4).

The experimental result showing the performance of those parsing strategies is illustrated in Table 4.6. The "unnamed constituent parsing" had the highest precision (84.43%) and F-score (85.40%) while the quality of the constituent parsing with grammar-functional labels was lower than that of the unnamed one about 14%. There was a small gap between the accuracy of the standard constituent parsing and the unnamed one (about 3% difference).

4.4 Preliminary proposal on syntactic rules and breaks

This is our first analysis and proposal on predicting break indices using syntactic information for Vietnamese TTS systems. It is believed that there is an interface between syntax and prosodic structure (Martin, 2010, Nespor and Vogel, 1983, 2007, Selkirk, 2011, 1980). However, in this preliminary study, we did not investigate much on the theory of prosodic syntax interface. Some syntactic rules were proposed for predicting levels based on the preliminary study on the theory, and mostly based on our observations on prosodic and syntactic structure. We assumed that break indices above word and below sentence were from "2" to "4" (cf. Section 5.4, Chapter 5 for other levels).

4.4.1 Proposal process

In this work, the VNSP-Broadcaster corpus, recorded by a broadcaster – an existing small one with 630 sentences – was used for analyses and proposal. As presented in the Section 4.2, in this corpus, audio files were manually transcribed, time-aligned at the syllable level, and annotated for perceived pauses in TextGrid files. Text files were manually parsed to constituent syntax trees with additional grammar-functional labels.

Main tasks for proposal of hypotheses with syntactic rules linked to relevant break indices are illustrated in Figure 4.4. Hypotheses including syntactic rules to predict prosodic boundaries with corresponding break indices were proposed based on our observation in corpora. These hypotheses were applied to syntactic trees of text corpus to specify prosodic boundaries, which were then automatically annotated into TextGrid files of audio corpus to identify prosodic phrases. Last syllables and next pauses of these predicted phrases were measured and analyzed. Durations of last syllables and next pauses of predicted phrases were measured. Final lengthening of last syllables was calculated based on Z-score normalization, linked to syllabic structures and tone types. Statistical analyses were carried out to find a correlation between syntactic element boundaries and pause duration as well as final lengthening. This

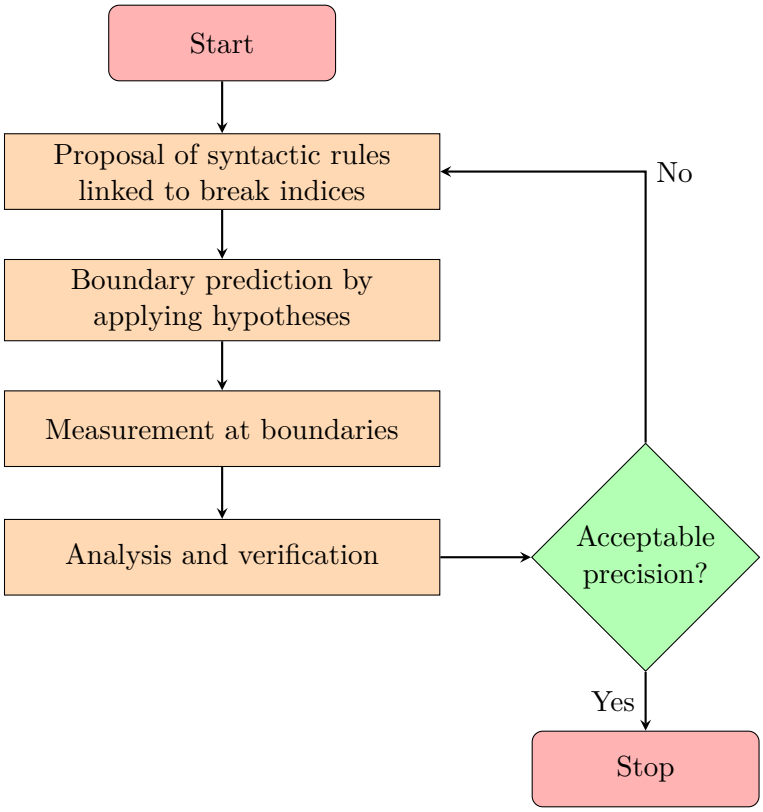


Figure 4.4 – General approach for prosodic phrasing modeling using syntactic rules.

process was repeated with new or fine-tuned syntactic rules until an acceptable precision of boundary prediction was obtained.

Constituents in phrase structure grammar or dependents in relational structure were used as primary elements of syntactic rules. break indices for these rules were proposed based on the possibility of pause appearance and pause length at predicted boundaries. After several iterations, we discovered some features for fine-tuning, i.e. number of syllables of dependents/constituents, children or parents of dependents/constituents.

4.4.2 Proposal of syntactic rules

Formal symbols were proposed to formally express syntactic rules for further automatic processing in boundary prediction and fine-tuning. A detail of these proposed symbols is presented in Appendix B.

After having studied the theory of syntax-prosody interface and observed relations between syntax and pause appearance in the corpus, we proposed some hypotheses on syntactic rules with corresponding break indices. Two types of rules were discovered: (i) Constituent syntactic rules between two constituents in phrase structure grammar and (ii) Functional rules between two dependents in relational structure. Proposed rules with syntactic constituents and with syntactic dependents are presented respectively in Table B.2 and Table B.3, Appendix B.

The highest break level in middle of sentences ("4") were set if either the left constituent is or contains a clause (S, SB), i.e. the rules HC1 and HC2, or both left and right dependent elements were predicates (PRD), i.e. the rule HD1, or head elements (H), i.e. the rule HD2.

Other decisions were made on the basis of syntactic element names (e.g. adjuncts *ADT*) or/and number of syllables in the left or right elements. Smaller break indices ("2" and "3") may appear after some special POS or syntactic phrases, e.g. prepositional phrases *PP*, conjunction *C*. Syntactic rules were refined using number of syllables, parents or children of syntactic elements. For instance, we found that there was a boundary between a phrase having at least 7 syllables, and a phrase having at least 4 syllables (HC3). These number limits were optimized through several iterations from proposal to evaluation.

For instance, the formal representation for the syntactic rule HC1 is "$SB;.\{1,\}(child : S|SB)-$". It means that there is a boundary between a subordinate clause (SB) or any constituents having a clause child ("$child : S|SB$") AND any constituent, such as "[Người đàn ông [mà bà gặp hôm qua ở nhà tôi]$_{SB}$]$_{NP}$ − là một người rất tốt bụng" *([The man [you met yesterday at my home]$_{SB}$]$_{NP}$ − is a really kind person)*. With the rule HC5: "$PP >= 3 - C; [ANV]P$", we assumed that there was a boundary between a prepositional phrase having at least 3 syllables ("$PP >= 3$") AND a conjunction ("C") or an adjective/noun/verb phrase ("$[ANV]P$"), such as "Đó là kết quả của những buồn vui [trong tình yêu của riêng mình]$_{PP}$ − [và]$_{C}$ cả những tâm sự của khán giả dành cho tôi" *(That is the results of joyfulness and sadness [in their own love]$_{PP}$ − [and]$_{C}$ also their confidings given to me)*.

For dependent rules, we only investigated on some typical cases. For example, there were usually a boundary between two predicates (HD1: "$PRD - PRD$"), e.g. "Lão [có nhà ở ngoại ô]$_{PRD}$ − [có ô tô hạng sang và vài người giúp việc]$_{PRD}$" *(He [had a house in a suburban area]$_{PRD}$ − [had a luxury car and several housekeepers]$_{PRD}$)*. Another instance of dependent rules we found is the rule HD5: "$2 <= ADT <= 3 - SUB >= 2$". It means that there is a boundary between an adjunct having 2 or 3 syllables ("$2 <= ADT <= 3$") AND a subject having at least 2 syllables ("$SUB >= 2$"), e.g. "[Đột nhiên]$_{R-ADT}$ − [lão]$_{NP-SUB}$ bán tất để chuyển vào thành phố" *([Suddenly]$_{R-ADT}$ − [he]$_{NP-SUB}$ sold everything in order to move to city)*.

A detail description of proposed syntactic rules with some examples shown in the two tables is presented in Appendix B.

4.4.3 Rule application and analysis

For further analysis and evaluation of the accuracy of the proposed syntactic rules, these rules were **automatically applied** into syntax trees to predict boundaries, which were then identified, and annotated into TextGrid files in a tier named by the rule (from HC1 to HC7, or from HD1 to HD5). Durations of last syllables right before and the ones of pauses right after predicted boundaries were measured in TextGrid files. Pause durations were computed in a logarithmic scale, which was more relevant to perception. Final lengthening is calculated using Z-score normalization, based on syllable structures and tone types of the last syllables.

This process is illustrated through an example in Figure 4.5, for the sentence "Do đó, khách nước ngoài đến Việt Nam tham gia các lễ hội thường thấy thú vị trước những tập tục này" *(Therefore, foreigners who come to Vietnam for participating traditional festivals find these customs interesting)*. All proposed syntactic rules including constituent rules (from HC1 to HC7) and dependent rules (from HD1 to HD5) were applied to the corresponding syntax tree. For this sentence, two rules were matched to predict two prosodic boundaries: (i) after the syllable "đó" (HD5 - there is a boundary between an adjunct ADT having 2 or 3 syllables and a subject SUB having at least 2 syllables) and (ii) after the syllable "hội" (HC1 - there is a boundary between a subordinate clause SB and any constituent).

As a result, 2 tiers naming HC1, HD5 were inserted into TextGrid files with respectively intervals naming prosodic phrases (Phrase). Durations of last syllables "đó" and "hội" of

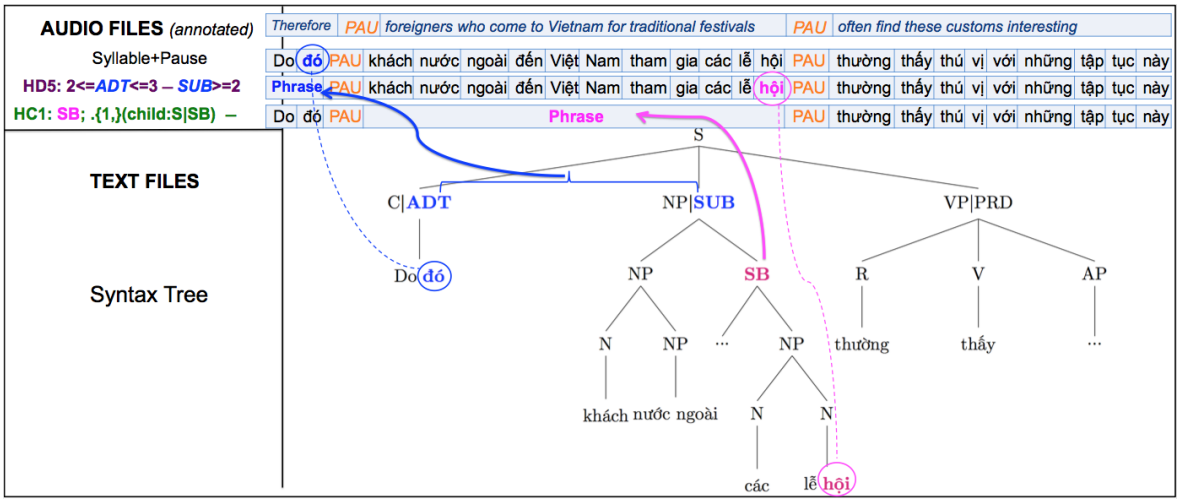


Figure 4.5 – An example of rule application to syntax tree and transcription file. This process was automatically performed by our program.

two predicted phrases were measured by ZScore normalization. Pauses succeeding these last syllables were also measured in *ms*, and computed in a logarithm scale. If there was no pause after these last syllables, pause durations after these syllables were set to 0 (called zero pauses).

Table 4.7 – ANOVA results of Syntactic Rules and break indices on Pause length and Final lengthening

Anova	df	df error	F	P	η^2
Pause∼Syntactic Rule	11	531	8.2	0.000	0.14
Log(Pause)∼Syntactic Rule	11	486	8.3	0.000	0.16
Lengthening∼Syntactic Rule	11	531	3.9	0.000	0.07
Pause∼Break Level	2	540	34.7	0.000	0.11
Log(Pause)∼Break Level	2	495	34.3	0.000	0.12
Lengthening∼Break Level	2	540	2.7	0.067	0.01

Analyses of variance were run on pauses lengths and final lengthening. The fixed factors considered in each ANOVA were "Syntactic Rule" (12 levels) and "Break Level" (3 levels). To eliminate the side effect of taking the logarithm of cases where pauses had a duration of zero (no pause), all zero pause cases (40/547) were removed from these analyses based on Log(Pause). Table 4.7 shows the ANOVA results. All analyses were significant ($p<0.05$), except the effect of Break Level on Lengthening ($p=0.067$). The results showed that the proposed syntactic rules and break indices were mainly related to the pause length.

Figure 4.6 illustrated the final lengthening and pause length (log scale) of predicted break indices (using syntactic rules). In this figure, we can see increases of Log(Pause) by the break indices, especially between the break level "2" and "3". The gap of Log(Pause) between the break level "3" and "4" is small but significant (as presented in Table 4.7, $p<0.05$). However, we did not find any rule of final lengthening on the break indices.

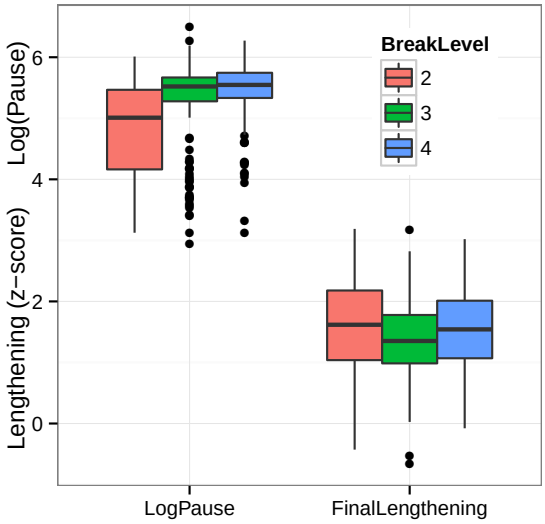


Figure 4.6 – Final lengthening (ZScore) and Log(Pause) of predicted break indices.

4.4.4 Evaluation of pause detection

The pause prediction performance of syntactic rules in manual environment (VNSP-Broadcaster) had a rather high precision, as illustrated in Table 4.8. Most of them could predict nearly 90% to 100%, except for the rules HC6 (82.9%) and HD5 (78.3%). These two rules were constrained by a small number of syllables for syntactic elements (3 to 5 for HC6 and 2 to 3 for HD5). The rule HC7 had a small observation number in this corpus since a specific conjunction "rằng" (that) was used in that rule.

Table 4.8 – Detail precisions syntactic rules in VNSP-Broadcaster and VTDO-Analysis

Break level	Rule code	VNSP-Broadcaster (manual)			VDTO-Analysis (automatic)		
		Pau/Total	Precision	Pau Mean	Pau/Total	Precision	Pau Mean
4	HD3	59/60	98.3%	253.24	7/7	100.0%	243.12
	HC2	75/79	94.9%	250.70	771/812	95.0%	281.35
	HD2	28/28	100.0%	266.89	2/3	66.7%	220.91
	HC1	26/27	96.3%	254.49	145/180	80.6%	236.81
	HD1	23/23	100.0%	248.25	32/32	100.0%	280.26
3	HC3	62/68	91.2%	240.92	391/504	77.6%	244.15
	HC4	48/54	88.9%	224.68	460/537	85.7%	173.74
	HC5	45/51	88.2%	203.56	230/370	62.2%	235.3
	HC7	08/08	100%	178.45	6/11	54.5%	184.71
	HC6	29/35	82.9%	160.51	351/383	91.6%	262.17
2	HD4	79/89	88.8%	147.06	165/219	75.3%	229.36
	HD5	18/23	78.3%	136.31	69/87	79.3%	202.42

In an automatic environment and a bigger corpus, VDTO-Analysis, precisions were divided into several groups. Since both constituent and dependent rules were proposed, and phrase names are necessary, the “Constituency parsing with functional labels” (cf. Section 4.3) was necessary. This type of parsing had the lowest precision (70.56%) and F-score (72.33%)

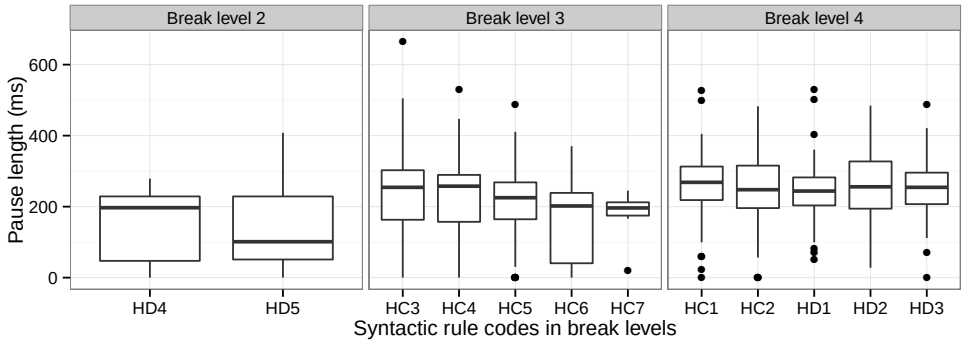
while the quality of the "unnamed constituency parsing" was higher than that of the complete one about 14%.

The rules HD3, HC2, HD1 and HC6 had a high precision (more than 90%). With these rules, boundaries were predicted after main clauses (HC2, HC6 with different number of syllables for the left syntactic elements) or between two equal elements (two head elements in HD3 or two predicates in HD1). Whereas, the rules HD2, HC5 and HD7 had worse precisions (i.e. lower than 70%). The reason was either their sparseness in the corpus or their effectiveness in automatic environment. The remaining ones were from 75.3% (HD4 between two head elements having from 2 to 3 syllables), about 80% (HD5 - after adjuncts with 2 or 3 syllables or HC1 - after a subordinate clauses) or 85.7% (HC4 - between two rather long syntactic elements: at least 7 syllables and at least 4 syllables).

Some rules had extremely small observation numbers, i.e. HD3 (7/7), HD2 (2/3), HC7 (6/11), HD1 (32/32) or rather small compared to VNSP-Broadcaster by size, i.e. HD5 (69/87), HD4 (165/219). They were all dependent rules (from HD1 to HD5) and only one constituent rule, which had a specific conjunction rằng *(that)* in a subordinate clause. Labels using in dependent rules had a low recall of syntax parsing.

Distributions of detected pause length by syntactic rules are illustrated in Figure 4.7, factored by relevant break indices for both corpora. In VNSP-Broadcaster, rules in the break level 2 had a smaller median and low separate from zero points (no pause) than the others. There was no considerable difference among syntactic rules in the break indices 3 and 4. In VDTO-Analysis, there was only one break level for all syntactic rules in automatic environment. We believed that in manual environment, it was necessary to do more analyses with a bigger and rule-balanced corpus.

(a) VNSP



(b) VDTO

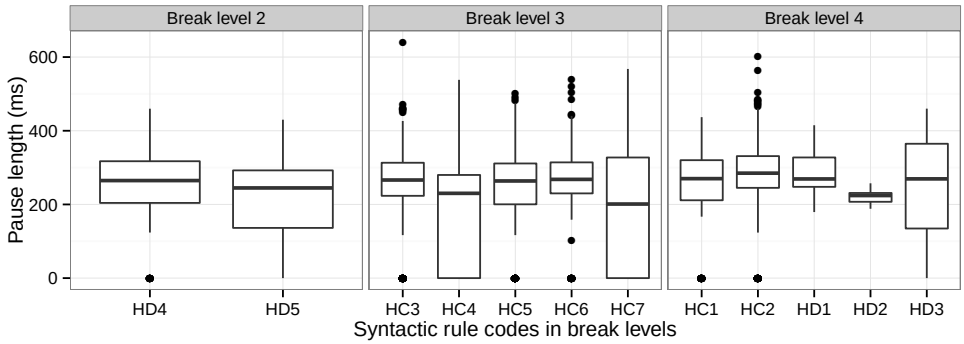


Figure 4.7 – Distributions of pause length of predicted boundaries by break indices using syntactic rules in (a) VNSP-Broadcaster (b) VDTO-Analysis.

Final evaluation of syntactic rules is presented in Table 4.9. In the VNSP-Broadcaster corpus, the precision of these rules applied in manually-parsing trees was high – 91.2% while the recall of them was rather low, 38.5%. The F-score was then 54.1% and $F_{0.5}$ was 71.6%. Whereas, in the automatic environment with the VDTO-Analysis corpus, the precision and recall were both lower about 10-12% than VNSP-Broadcaster. Hence the F-score reduced about 13.7% to 40.4% and $F_{0.5}$ was 58.4%.

Table 4.9 – Evaluation of syntactic rules in VNSP-Broadcaster and VTDO-Analysis corpora

Corpus		**Pause #**	**Correct pau #**	**Predicted pau #**	**P**	**R**	**F**	$F_{0.5}$	F_2
VNSP (manual)	HC1-7	790	293	322	91.0%	37.1%	52.7%	70.5%	42.1%
	HD1-5		207	223	92.8%	26.2%	40.9%	61.5%	30.6%
	Both		**425**	**466**	**91.2%**	**53.8%**	**67.7%**	**80.1%**	**58.6%**
VTDO (auto)	HC1-7	9,005	2,354	2,797	84.2%	26.1%	39.9%	58.3%	30.3%
	HD1-5		275	348	79.0%	3.1%	5.9%	13.2%	3.8%
	Both		**2,408**	**2,905**	**82.9%**	**26.7%**	**40.4%**	**58.4%**	**30.9%**

It turned out that the prosodic phrasing model using syntactic rules considerably depended on the quality of syntax parser. The quality of the adopted syntax parser worked well for constituent parsing (with rules HC-HC7), but poorly generated dependent elements (with rules HD1-5). The quality of the prosodic phrasing model in the automatic environment was mainly based on the constituent parsing as the recall of dependent rules was not considerable (3.1%).

4.5 Simple prosodic phrasing model using syntactic blocks

The preliminary study showed that syntactic rules could provide a good prediction with manual parse trees, but only constituent syntactic rules gave a good precision (P=84.2%), but poor recall (F-score=39.9%) in the automatic environment. Furthermore, it is necessary to have a thorough study on theory of prosody syntax interface for further investigation.

In this section, another approach using syntactic trees for prosodic phrasing using durational clues alone was proposed. In this part of work, with a motivation of automatic TTS, the VDTO corpus (cf. Section 4.2) was considered. The VDTO-Analysis corpus with nearly 5,000 sentences, 10 times larger than the old one VNSP, was used in the analysis. Audio files in this corpus were automatically segmented, time-aligned at phoneme level and perceived pauses while text files are automatically parsed into XML files by VTParser. Based on the results of our analysis, only the "Unnamed constituent parsing", whose performance was best, was needed (cf. Section 4.3).

As aforesaid, Vietnamese text is a sequence of syllables, separated by spaces; and there are more than 80% single-syllable words. Therefore, syllables in continuous speech are quite independent and crucial units composing utterances. We hence investigated on studying the relation between syllable durations and prosodic phrasing. A syllable duration pattern of different levels of syllables' ancestors was discovered to be similar to that of breath groups, which includes syllables between two consecutive perceived pauses, one of the most prominent and frequent levels of prosodic phrasing (Parlikar, 2013). Syntactic blocks – syllable ancestors with bounded size – were finally proposed for two levels of prosodic phrasing: (i) final lengthening, and (ii) pause appearance.

4.5.1 Duration patterns of breath groups

As presented, one of the most prominent and frequent phrase break levels is a pause. We hence first investigated on syllables in two consecutive perceived pauses, i.e. *breath groups*, in the VDTO-Analysis corpus. The maximal syllable number of breath groups in the VDTO-Analysis corpus was 27 (only one utterance), and few utterances ranged from 24 to 26 syllables (see Figure B.3 in Appendix B for the distribution of breath group length). We were interested in duration of syllables by their positions in breath groups having the same size (e.g. 6 syllables).

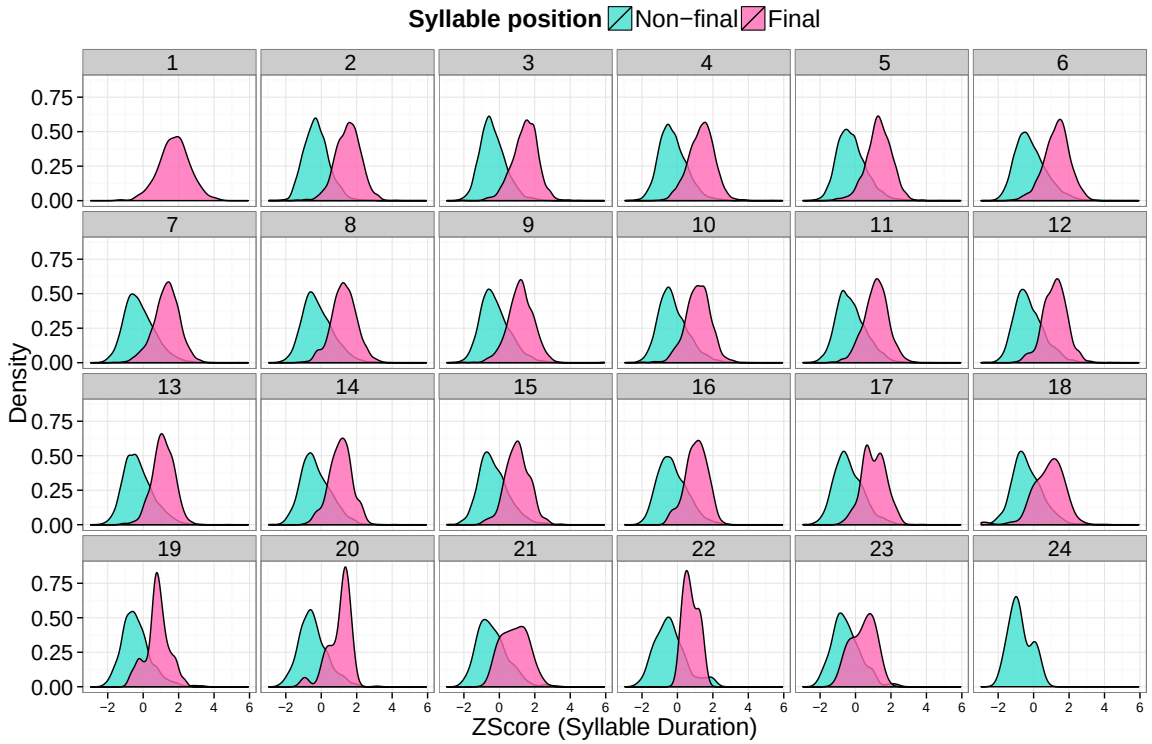


Figure 4.8 – Distributions of non-final/final syllable duration (ZScore) of breath groups in the VDTO-Analysis corpus, factored by syllable numbers of breath groups. Breath groups having more than 24 syllables were excluded.

Based on the observation on the corpus, we assumed that the duration of the final syllable in a breath group played an important role in the prosodic phrasing of that breath group. Figure 4.8 illustrates the distributions of non-final/final syllable duration (represented by ZScore) of breath groups in the VDTO-Analysis corpus, factored by syllable numbers of breath groups. In this figure, breath groups having more than 24 syllables were excluded due to their sparse observation in the corpus. This figure shows that there was a remarkable lengthening of final syllables over non-final syllables in spite of syllable numbers of breath groups. This final lengthening could be apparently observed in breath groups with less than 18 syllables due to their sufficient observation numbers.

Syllable positions in breath groups were then investigated. Figure 4.9 illustrates the distributions of syllable durations (ZScore) by syllable positions in breath groups in the VTDO-Analysis corpus, factored by syllable numbers of breath groups. For ease of observation, from now on, this plot type excluded the groups with more than 18 syllables. In this figure, level

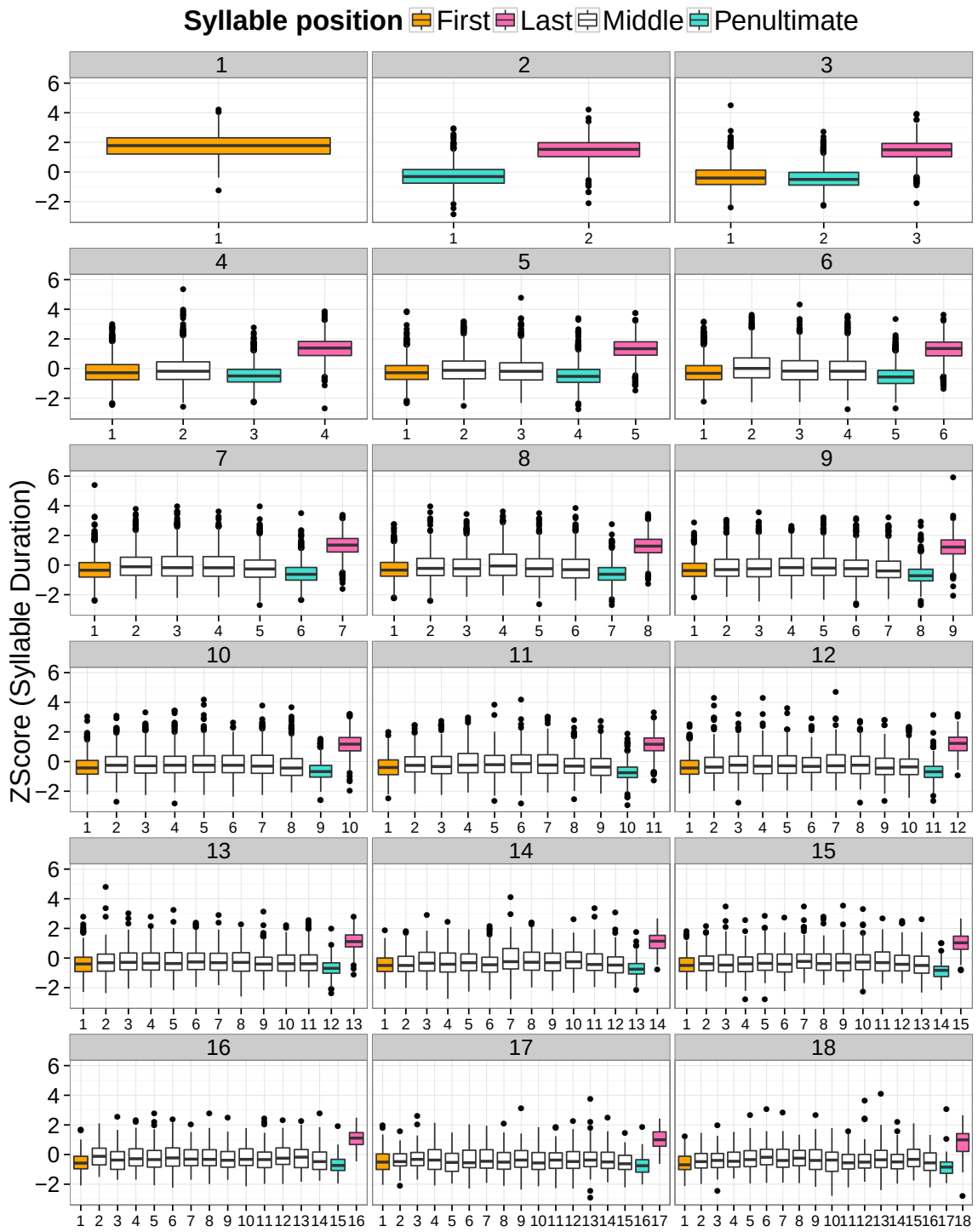


Figure 4.9 – Distributions of syllable durations (ZScore) by syllable positions in breath groups in the VTDO-Analysis corpus, factored by syllable numbers of breath groups. Breath groups having more than 18 syllables were excluded.

of final lengthening of shorter breath groups was slightly higher than the longer ones, but not considerable. There was a shortening in the penultimate syllables and a slight shortening in first syllables. Normalized duration of middle syllables varied insignificantly. We found that the pattern of normalized duration of syllables in breath groups was follows: *Slight shortening in the first syllable, no special variation in middle syllables, shortening in the penultimate syllable and final lengthening in the last syllable.*

4.5.2 Duration pattern of syllable ancestors

Syllable ancestors. We assumed that there was a relation between syntactic phrases and breath groups in terms of syllable duration. Syntactic phrases in this work were any ancestors of words (i.e. above words) in hierarchical syntactic trees. They included both grammatical phrases (cf. Section 4.3), and clauses. We then investigated into syntactic phrases that are highest and lowest level ancestors of bearing words of examined syllables.

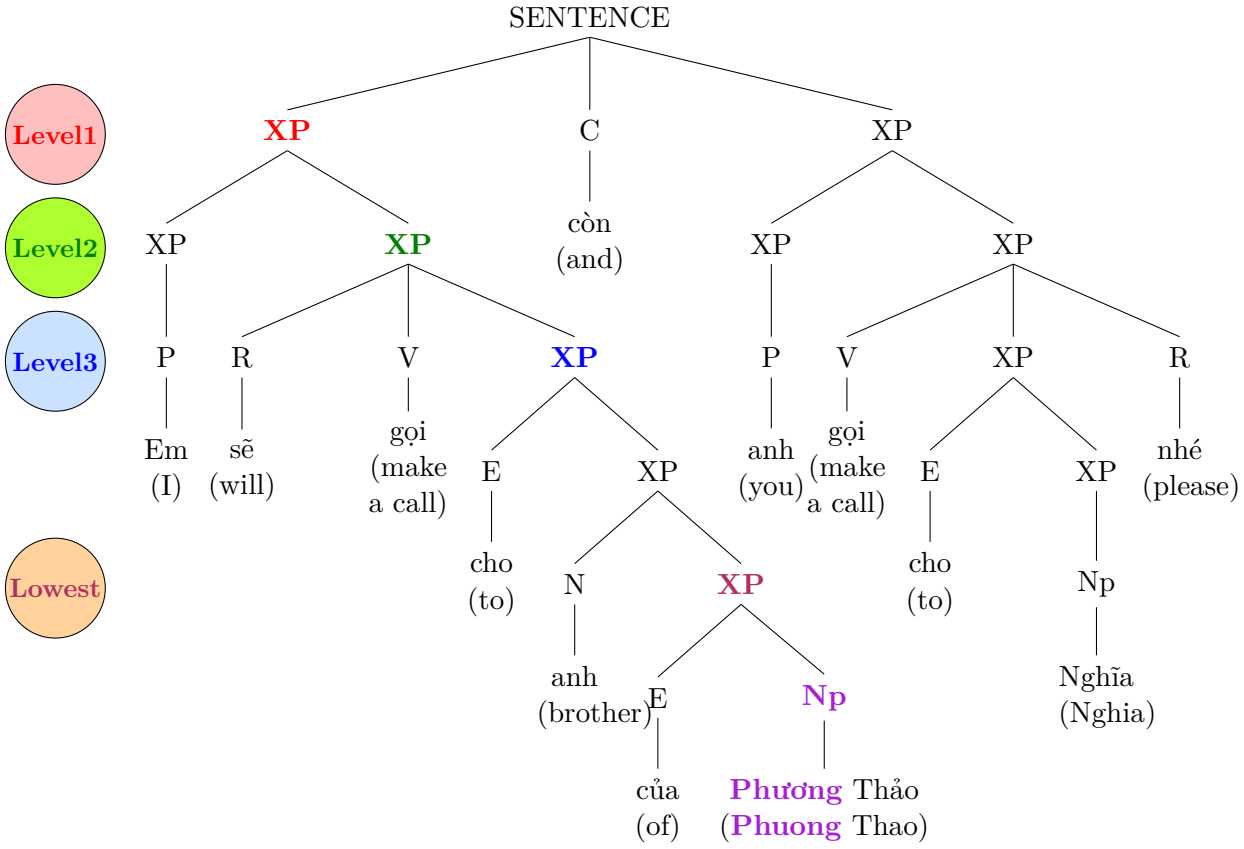


Figure 4.10 – Highest (1st, 2nd and 3rd level) and lowest ancestors of the syllable "Phương" *(Phuong)* in syntax tree.

In general, the highest ancestor could be either the main clauses of compound sentences, or whole simple or complex sentences. Ancestors of one syllable were the ancestors of the bearing word. For instance, Figure 4.10 shows an example of a compound sentence "Em sẽ gọi cho anh của Phương Thảo còn anh gọi cho Nghĩa nhé!" *(I will make a call to Phuong Thao's brother and you make a call to Nghia, please!)*. In this figure, highest and lowest ancestors of the syllable "Phương" are shown. The ancestors of the syllable "Phương" were

ancestors of the bearing word "Phương Thảo" (proper noun Np).

The lowest ancestor of the word "Phương Thảo" was its parent, the phrase "của Phương Thảo" *(of Phuong Thao)*. The highest level (first level) ancestor of the word "Phương Thảo" *Phuong Thao* was its ancestor, one step lower than the root node "SENTENCE", i.e. the clause "Em sẽ gọi cho anh của Phương Thảo" *(I will make a call to Phuong Thao's brother)*. The second level ancestor of that word was its ancestor, one step lower than the highest level ancestor, i.e. the phrase "sẽ gọi cho anh của Phương Thảo" *(will call to Phuong Thao's brother)*. Similarly, the third level ancestor was the phrase "cho anh của Phương Thảo" *(to Phuong Thao's brother)*.

Syllable duration patterns. Ancestors of all syllables in the VDTO corpus at the first level (highest ancestor), second level, third level and the lowest level (lowest ancestor) were automatically extracted using syntactic trees for analysis. We found that the duration pattern by syllable positions in all levels of syllable ancestors was similar to that of breath groups. Figure B.4 illustrates the distributions of syllable-normalized duration by positions in lowest ancestors (see Figure 4.11 in the Appendix B for the highest ancestor). In the same size of syllable ancestors, the first syllable was slightly shortening, the penultimate one was shortening, and the last one was lengthening. This finding gave us a hypothesis of an important role of these syntactic phrases in prosodic phrasing modeling, at least for predicting pause appearance.

Apart from last syllables of utterances that had a subsequent pause and a big degree of final lengthening, about 30% to 43% last syllables of different level ancestors had subsequent perceived pauses (i.e. last syllables of breath groups). We supposed that those last syllables of breath groups might contribute final lengthening of syllable ancestors. To have a better illustration for this assumption, all syllables having subsequent pauses (including last syllables of utterances) were excluded. Furthermore, to remove the impact of higher levels, last syllables of all higher-level ancestors of a certain level were also excluded for further analysis.

We first did this analysis for the second level ancestor with a condition: all last syllables of the highest ancestor and breath groups were removed. Figure 4.12 illustrates the distributions of those syllable durations (represented by ZScore normalization) by positions in the second level ancestors, factored by syllable numbers of these ancestors. The plot shows that even with ancestors without subsequent perceived pauses and last syllables of all higher-level ancestors, the same pattern still appeared with a lower degree of final lengthening. However, the shortening of the penultimate syllable and the lengthening of the last syllable in syllable ancestors were obviously observed.

We did more investigation and found that this pattern was expressed better by duration differences of two consecutive syllables. Figure B.5 in Appendix B illustrates the same distribution but of syllable duration differences by positions in the third level ancestor, excluding all last syllables of breath groups and the ones of two-higher-level ancestors. The difference between the penultimate and ultimate syllables was bigger.

From all above analysis on different level ancestors of syllables, some findings were proposed:

1. The pattern of syllable duration (represented by ZScore normalization) in any level of syllable ancestors having at least two syllables was the same as the one in breath groups: (i) slightly shortening at the first syllable and significantly shortening at the penultimate one, and (ii) strong degree of lengthening at the last syllable. We assumed that syllable ancestors were valuable candidates for predicting prosodic phrasing.

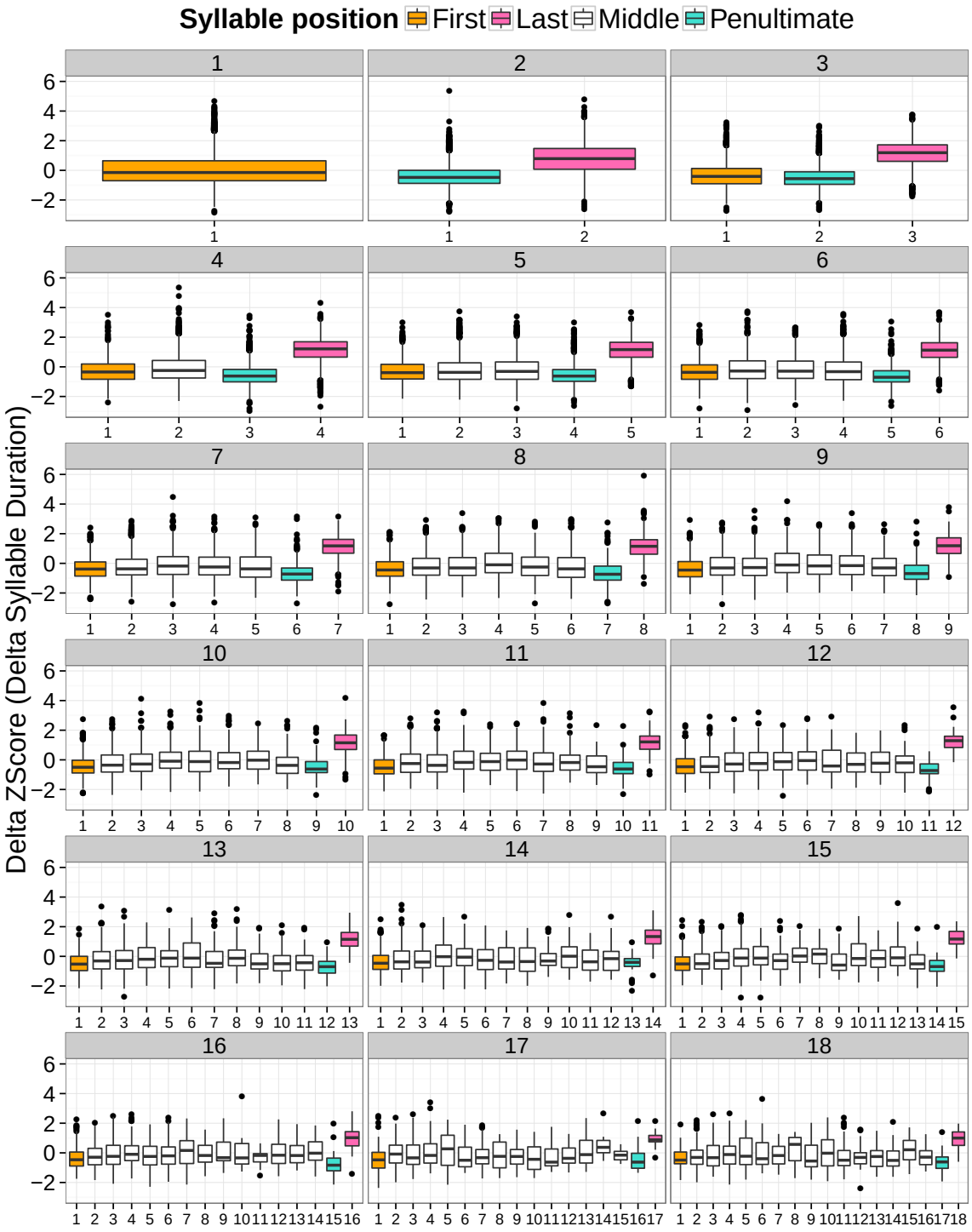


Figure 4.11 – Distributions of syllable durations (ZScore) by syllable positions in lowest ancestors in the VTDO-Analysis corpus, factored by syllable numbers of these ancestors. Breath groups having more than 18 syllables were excluded.

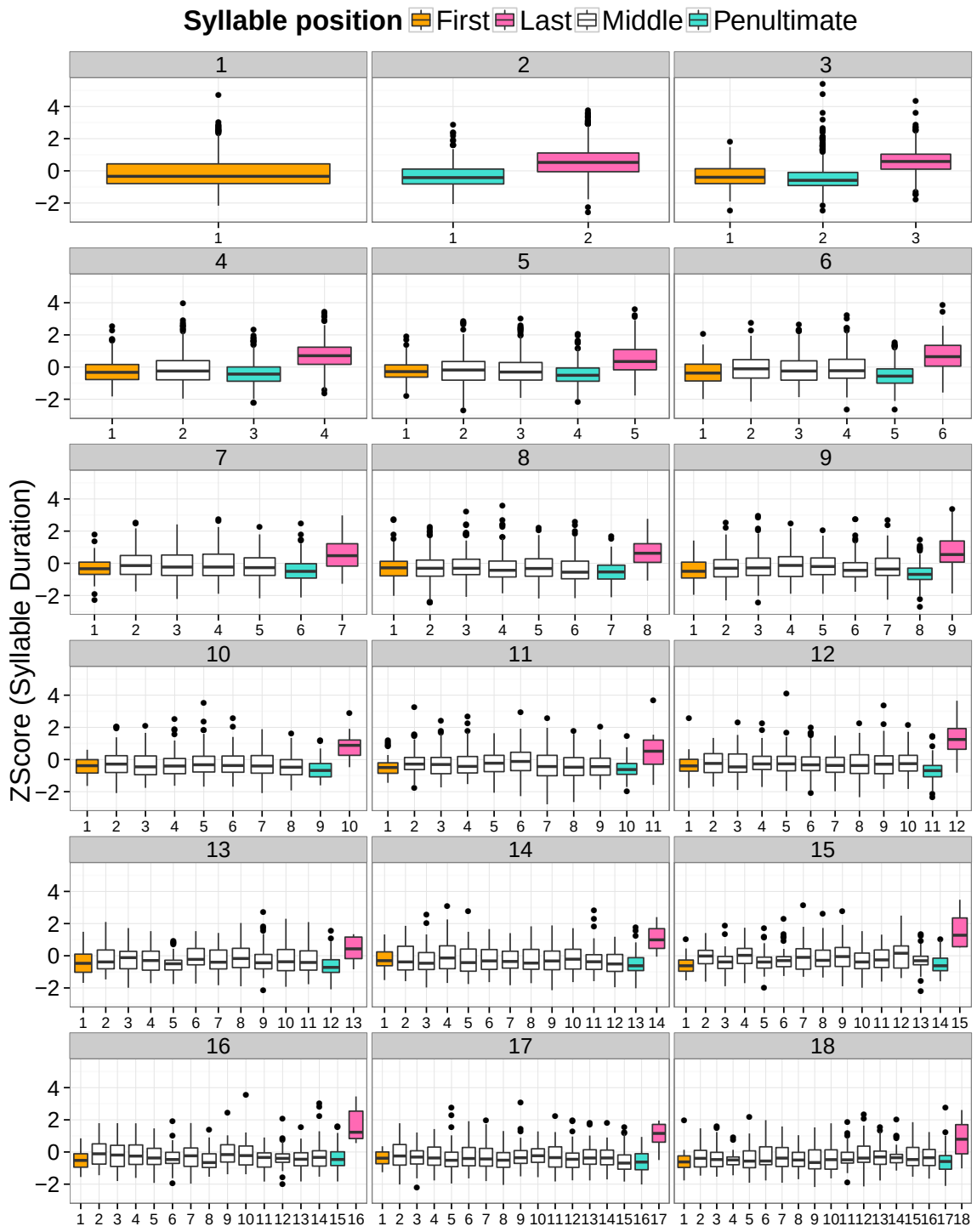


Figure 4.12 – Distributions of syllable durations (ZScore) by syllable positions in second level ancestors, factored by syllable numbers of these ancestors. Last syllables of higher-level ancestors and syllables with subsequent pauses were excluded. Ancestors having more than 18 syllables were excluded.

2. Final lengthening existed with a lower degree even in the last syllables of ancestors excluding last syllables of higher-level ancestors and breath groups. As a result, we supposed that there were two levels of prosodic phrasing using durational clues alone: (i) pause appearance and (ii) final lengthening.

4.5.3 Proposal of syntactic blocks

As presented in the previous subsection, syllable ancestors can be an effective cue to predict prosodic boundaries. However, syllable numbers of these ancestors varied from 1 to 70 syllables due to the structure and complexity of sentences. Table 4.10 shows the summarization of syllables numbers of lexical words, breath groups and different level ancestors of syllables (the first to the third and the lowest ancestors). The syllable numbers of one ancestor (even with the lowest-level one) were still 59 in many cases.

As observed, the duration pattern found in syllable ancestors was stable if their sizes were not exceed 24 (even with breath groups). We hence proposed that the levels of ancestors should be flexible, and the size of these ancestors (i.e. number of syllables) should be bounded. These "bounded" syntactic phrases were called syntactic blocks.

This turned to another problem of how to divide a sentence to syntactic blocks using syntactic trees. The solution for this problem can be follows. Let n was the bounded size of syntactic blocks. We extracted the first children of the root node. If any of them had more than n syllables, we kept dividing them to lower children until syllable numbers of all syntactic blocks were not above n. The main issue now was that how to figure out an optimized value for n, a maximal syllable number of all syntactic blocks. It can be induced that n should not be greater than the limit of breath group length in the corpus, i.e. 27 syllables.

Table 4.10 – Summarization of syllable number of breath groups and different level ancestors

Ancestor type	Maximal length	Mean length	Median length	3rd Quartile
Highest ancestors (1st level)	70	29.6	27	40
Pre-highest ancestors (2nd level)	64	17.1	14	25
Pre-Pre-highest ancestors (3rd level)	59	10.9	7	15
Lowest ancestors (lowest level)	59	6.9	5	9
Breath groups	27	8.6	8	11
Words	6	1.5	1	2

Figure B.6 in Appendix B shows the distributions of syllable duration by positions in syntactic blocks with a maximum of 27 syllables, factored by syllable numbers of these blocks. In this figure, all syllables with succeeding pauses were eliminated. The same syllable duration pattern can be observed in each syntactic block. However, the final lengthening of syntactic blocks having above 17 syllables was rarely observed or not stable. Therefore, we supposed that the maximal syllable number n of syntactic block should not be more than 17 syllables. Moreover, one syntactic phrase should contain at least one word, which had maximal length of 6 syllables in the corpus. There were only two observations for that maximal length of words: (i) loan word, i.e. "a-lếch-xan-đờ-rô-vích" *(a Russian person name)* and (ii) a long word, i.e. "phương tiện giao thông vận tải" *(transport traffic vehicle)*. Consequently n should be at least 6 syllables and not greater than 17 syllables.

We first did an analysis on syntactic blocks with a maximum of 17 syllables, the largest possible value for n. Figure 4.13 depicts the distributions of syllable duration (ZScore) and

subsequent pause length (log scale) of final syllables of these syntactic blocks, factored by their syllable numbers. If there was no subsequent pause, log(pause) was set to 0 for ease of representation. It was clear that after syntactic blocks having at least 2 syllables, there was a final lengthening, one cue of prosodic phrasing. The level of lengthening was higher at the end of syntactic blocks with at least 3 syllables. For another stronger cue of prosodic phrasing, i.e. pause appearance; we examined perceived pauses in the middle of utterances after syntactic blocks. Pause presence could be roughly predicted by syntactic block size: pauses mostly appeared at the end of syntactic blocks having at least 5 syllables, as shown in Figure 4.13. Based on the median of pause length in the figure, we supposed only one level of pause length. As a result, we proposed here two levels of prosodic phrasing for the Vietnamese TTS, i.e. final lengthening and pause appearance.

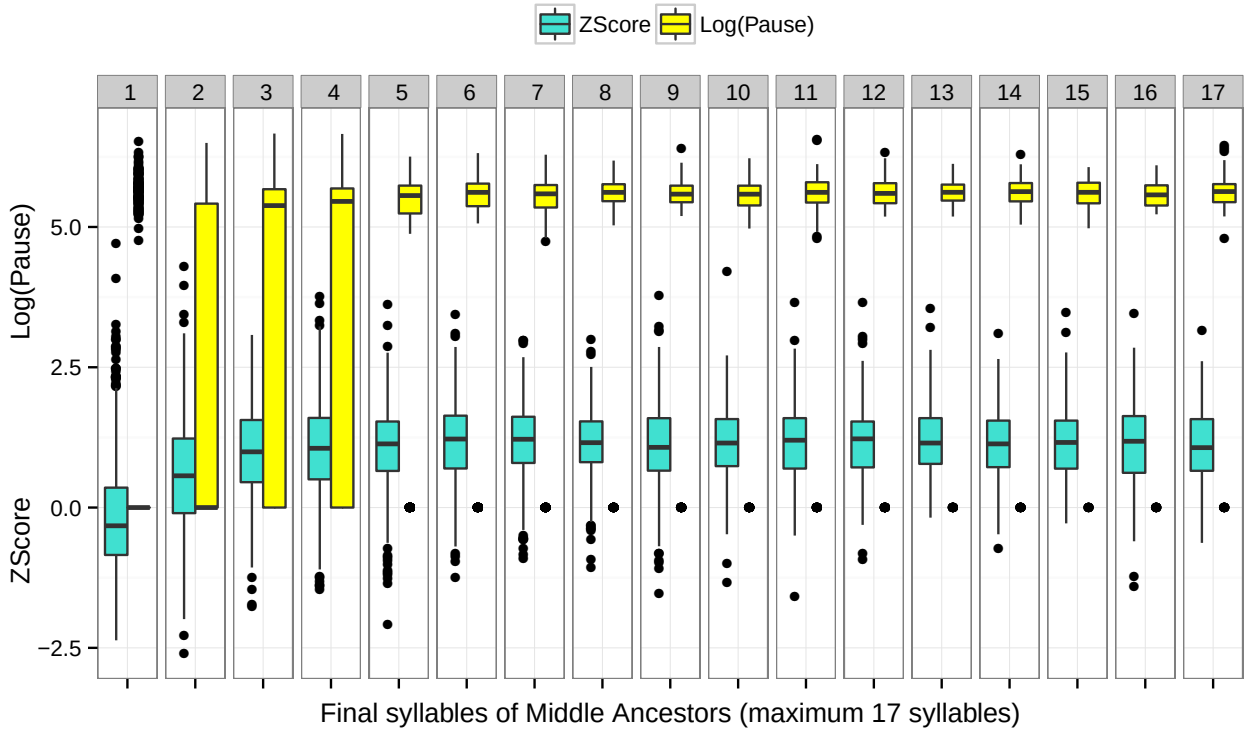


Figure 4.13 – Distributions of duration (ZScore) and subsequent pause length (log scale) of final syllables of syntactic blocks with a maximum of 17 syllables, factored by syllable numbers of these blocks. If there was no subsequent pause, log(pause) was set to 0 for ease of representation.

4.5.4 Optimization of syntactic block size

Based on the above initial idea for prosodic phrasing modeling, we present hereafter the optimization process to find out the value for n, the bounded value for syllable numbers of syntactic blocks, for predicting both final lengthening and pause appearance. We assumed that the criterion for final lengthening was to maximize the number of predicted lengthening; whereas the one for pause cue was to enlarge the accuracy and effectiveness of the prediction. In this optimization process, we adopted several metrics for evaluating the performance of

pause prediction: Precision, Recall and F-score (cf. Section 4.2).

Based on the above analysis of syntactic block length, we investigated into syntactic blocks with a maximum of syllable numbers from 6 to 17 syllables. As presented in Section 4.2, a total of 9,005 pauses inside utterances in the VDTO-corpus were found. Table 4.11 presents our measurements in different maximal syllable number of syntactic blocks, n from 6 to 17 syllables and the maximal length of breath group 27. It was easy to observe that when n increased, the number of predicted final lengthening was reduced. Therefore, the number of predicted lengthening was maximal when n was smallest (n=6, number of predicted final lengthening = 20,337). Compared to the total number of syllables in the corpus (89,717), about 23% of syllables were predicted as final lengthening syllables. This number (n=6) hence ultimately was chosen as the maximal syllable number of syllables for syntactic blocks to predict the final lengthening.

Table 4.11 – Different limits for syntactic blocks (n=6..17;27)

Bounded size (n)	**Predicted lengthening #**	**Correct predicted pauses #**	**Predicted pauses #**	**Precision**	**Recall**	**F-score**
6	**20,337**	3,097	3,884	79.7%	34.4%	48.1%
7	18,808	3,527	4,429	79.6%	39.2%	52.5%
8	17,452	3,722	4,623	80.5%	41.3%	54.6%
9	16,346	3,791	4,670	81.2%	42.1%	55.4%
10	**15,396**	**3,790**	**4609**	**82.2%**	**42.1%**	**55.7%**
11	14,559	3,711	4,482	82.8%	41.2%	55.0%
12	13,791	3,620	4,331	83.6%	40.2%	54.3%
13	13,101	3,500	4,164	84.1%	38.9%	53.2%
14	12,511	3,375	4,013	84.1%	37.5%	51.9%
15	11,949	3,257	3,843	84.8%	36.2%	50.7%
16	11,407	3,105	3,642	85.3%	34.5%	49.1%
17	10,971	2,977	3,482	85.5%	33.1%	47.7%
27	7,891	1,679	1,930	87.0%	18.6%	30.7%

To evaluate the performance of pause appearance, precision, recall and F-measure were calculated, as illustrated in Figure 4.11. When the bounded syllable number of syntactic blocks size increased, the Precision of the prediction gradually increased but the Recall reached the peak at the middle (n=9). As depiction in the table, when n=10, all measurements of F-score and both F_2, $F_{0.5}$ (these two metrics were not shown in the table due to the space), were maximal. We finally decided to choose the maximal syllable number of syntactic blocks was 10 for pause prediction though there were two other considerable candidates: n=9 or n=11.

4.5.5 Simple model for final lengthening and pause prediction

Based on our above analyses, we proposed a simple but effective model to predict two levels of prosodic phrasing: (i) final lengthening triggered at the last syllable of syntactic blocks (bounded syllable number=6) having at least two syllables (Figure 4.14) and (ii) pause appearing after syntactic blocks (bounded syllable number=10) having at least 5 syllables (Figure 4.15).

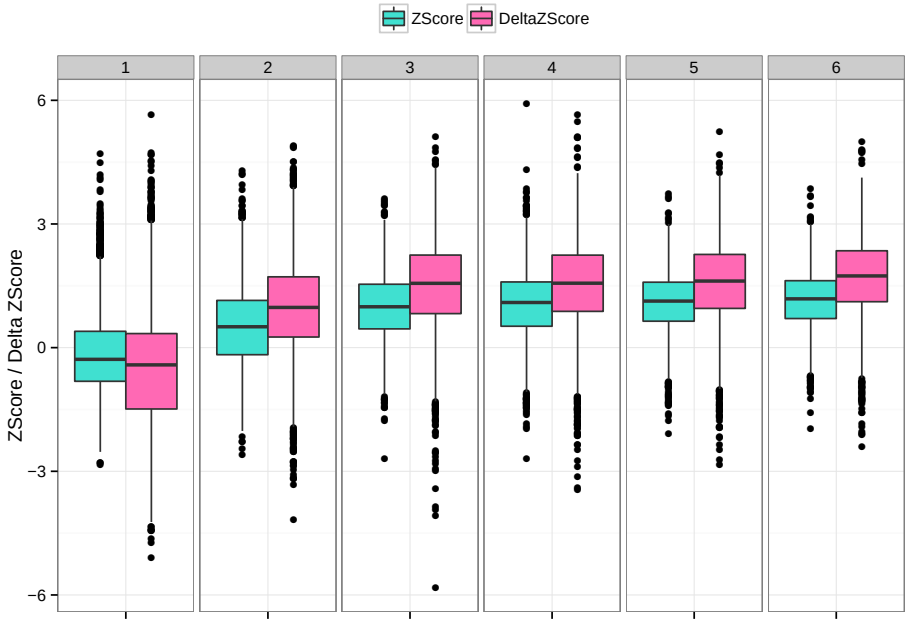


Figure 4.14 – Distributions of duration (ZScore normalization) of final syllables of syntactic blocks with a maximum of 6 syllables, factored by syllable numbers of these blocks.

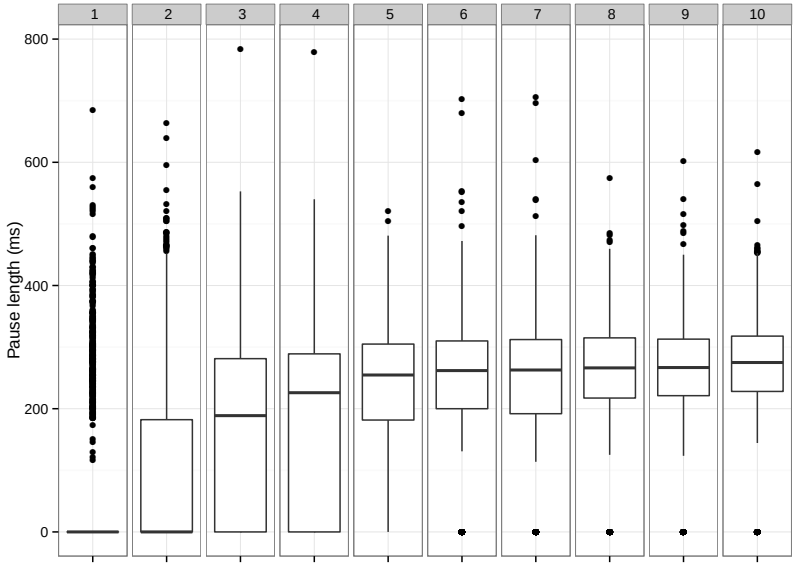


Figure 4.15 – Distributions of pause length of final syllables of syntactic blocks with a maximum of 10 syllables, factored by syllable numbers of these blocks.

Figure 4.14 illustrates the distributions of duration (ZScore normalization) of final syllables of syntactic blocks with a maximum of 6 syllables, factored by syllable numbers of these blocks. Final lengthening happened at the end of multiple-syllable syntactic blocks. However, the degree of final lengthening was stronger in the last syllables of syntactic blocks having more than 2 syllables. Figure 4.15 obviously shows that there were most pauses at the end of at least-5-syllable syntactic blocks with a maximum of syllable number of 10. The duration of

the last syllables of these blocks had a highly-separated distribution from zero (no subsequent pause).

Full algorithm for dividing one syntactic node (ancestor) to functional blocks with a limit number of syllables (*limitLength*) is presented in Listing 4.1. An ancestor was kept dividing into smaller dependent phrases until syllable numbers of all dependent phrases were bounded to *limitLength*. Recursion is used as the solution for this problem. To get syntactic blocks for a sentence, the root of that sentence and the maximal syllable number of syntactic blocks (*limitLength* = 6 for final lengthening prediction, *limitLength*=10 for pause prediction) are passed to the procedure.

In this procedure, the syllable number of the original node (ancestor) was first obstruent-final if it exceeds the *limitLength*. If not, that ancestor node would be returned. Otherwise, all direct children of that ancestor node were extracted. Each child of that ancestor node was passed with *limitLength* to the same procedure, whose results as syntactic blocks were added to the global return value of the original ancestor. This return value was a list of functional blocks (with a maximum of syllable number of *limitLength*) of the original ancestor node.

Listing 4.1 – Algorithm of syntactic block identification with a limit number of syllables

```
/**
     * Get list of descendant syntactic blocks whose maximal syllable number is
         limitLength
     * Each syntactic phrases include list of descendant words List<Node>
     * @RETURN phrases
               list of syntactic blocks List<Phrase>
     * @PARAM ancestor
               the ancestor node to get descendant syntactic blocks
     * @PARAM limitLength
               the limit syllable number of descendant syntactic blocks
*/
List<Phrase> getSyntacticBlocks(Node ancestor, int limitLength){
   INITIALIZE results AS a List<Phrase>
   IF syllable number of ancestor <= limitLength {
      RETURN ancestor;
   } ELSE {
      FOR EACH child OF ancestor {
         phrasesOfChild = getSyntacticBlocks(child, limitLength);
         ADD phrasesOfChild TO results;
      }
   }

   return results;
}
```

Since only the size of syntactic blocks and the positions of constituent syllables were considered, only the "unnamed constituency parsing" (cf. Section 4.3) is necessary. This type of parsing had the highest precision (84.43%) and F-score (85.40%) while the quality of the complete parsing with both phrase structure and functional labels was lower than that of the unnamed one about 14%.

4.6 Single-syllable-block-grouping model for final lengthening

4.6.1 Issue with single syllable blocks

As presented in the simple model, the final lengthening was predicted by multiple syllable syntactic blocks with a maximum of syllable number of 6. However, re-referencing Figure 4.14, the final lengthening still existed at the end of syntactic blocks having one syllable (single syllable syntactic blocks). For instance, if a single syllable syntactic block A was the last one in the utterance, the previous block might not have predicted well a final lengthening, but the last one A could. If there were several consecutive single syllable blocks, the last one could have predicted a final lengthening.

As a result, there was a need to do more investigation to discover additional points that could trigger final lengthening. Some cases supposed to have final lengthening might be removed if necessary.

4.6.2 Combination of single syllable blocks

The simple model for final lengthening could be improved by several strategies for combination of single syllable syntactic blocks. These strategies, drawn from our observations in the corpus for these syntactic blocks, are summarized as follows.

1. **ST-1**: A single syllable syntactic block X could only be combined with the next block (except some exceptions in the last strategy *ST-3*) to a new syntactic block if the previous block of X was a multiple-syllable block since this multiple-syllable block already had final lengthening. Several instances are shown in Figure 4.16. The previous block of the single block B had 3 syllables ($A-1$, $A-2$, $A-3$), hence the block B was combined with the next block C to a 3-syllable block (B, $C-1$, $C-2$). Similarly the single syllable block D was combined with the block E to a 4-syllable block (E, $E-1$, $E-2$, $E-3$) due to a previous 2-syllable block ($C-1$, $C-2$).

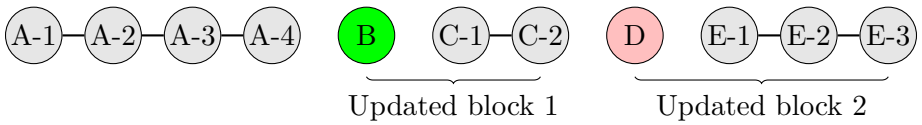


Figure 4.16 – Examples of combining single syllable syntactic blocks with the next block.

2. **ST-2**: A single syllable syntactic block X could only be combined with the previous block Y (except some exceptions in the last strategy) to a new syntactic block if the block Y have a single syllable. Several instances are shown in Figure 4.17. The single syllable block B was combined with the previous block A to a newborn syntactic block (A, B) because the previous block of C had only one syllables. The single syllable block E could not be combined with the next block F because the previous block D had only one syllable. Hence, block E was combined with the block D to become to a newborn syntactic block (D, E).

3. **ST-3**: Two exceptions for combining single syntactic blocks were: (i) all consecutive single-syllable syntactic blocks could be combined together as we do not know where to split them (ii) if the single syntactic block X was the last syllable of utterance, it could be combined with the previous block Y of X despite of the syllable number of the block

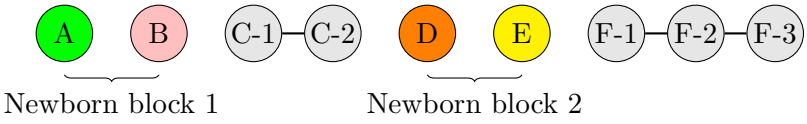


Figure 4.17 – Examples of combining single syllable syntactic blocks with the previous block.

Y (single or multiple). Several instances are shown in Figure 4.18. In Figure 4.18(a), all consecutive single syllable blocks A, B, C were combined to a newborn block. The single syllable block F was the last block of the utterance, hence it was combined with the previous block E although the block E had 3 syllables. In Figure 4.18(b), all 4 single syllable blocks C, D, E and F were combined to only one 4-syllable newborn block. The last block I of utterance was combined with the previous single syllable block H to a newborn block (H, I).

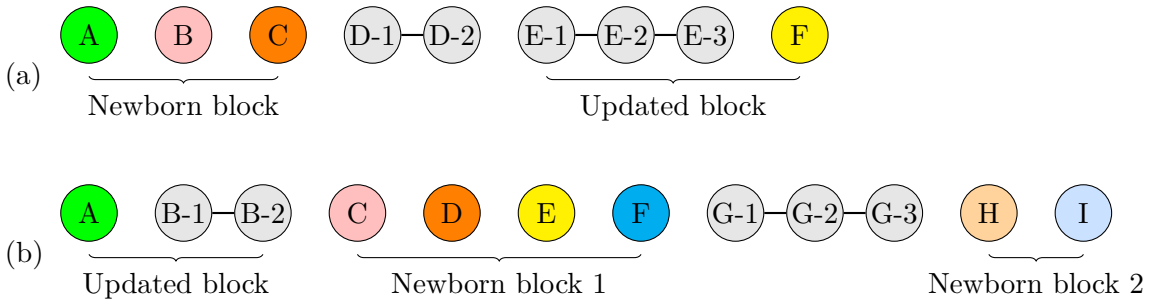


Figure 4.18 – Exception cases for combining single syllable syntactic blocks.

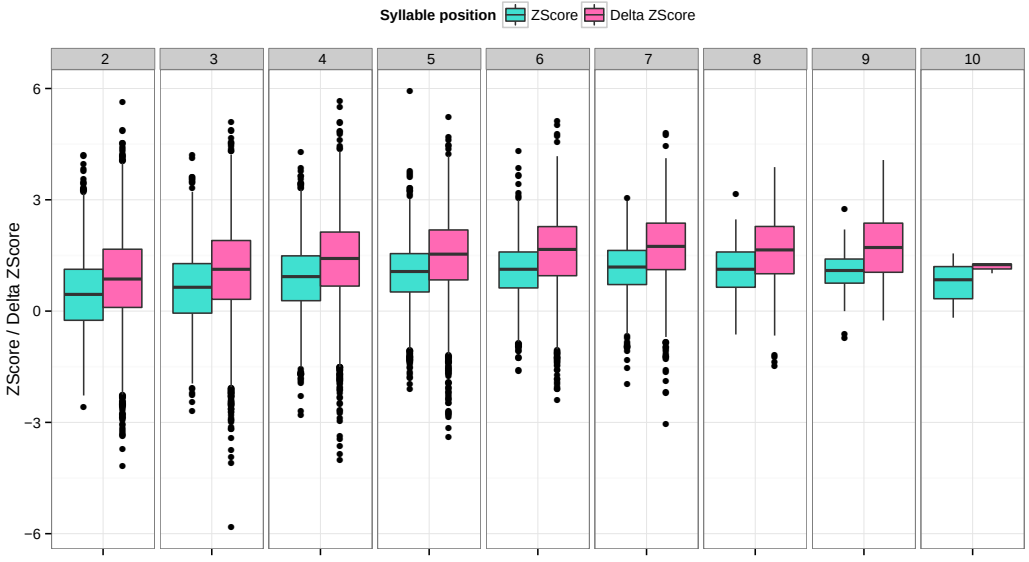


Figure 4.19 – Distributions of normalized duration (ZScore) of final syllables of combined syntactic blocks with a maximum of syllable number of 6, factored by syllable number of these blocks.

Figure 4.19 presents the duration distributions of final syllables of combined syntactic blocks. With the combination, there was no single-syllable block, and the final lengthening appeared systematically at the last syllables of all syntactic blocks. The degree of final lengthening of syntactic blocks having more than 2 syllables was still stronger than the one of blocks having 2 syllables. The number of final lengthening was increased about 16% from 20,337 to 23,683 positions.

Full algorithm for dividing one syntactic node (ancestor) to multiple syllable syntactic blocks (with some combination strategies of single syllable syntactic blocks) with a limit number of syllables (limitLength= 6 for lengthening) is presented in Listing B.1 in Appendix B.

4.7 Syntactic-block+link+POS model for pause prediction

As presented in the simple model, pause appearance was predicted by at-least-5-syllable syntactic blocks with a maximum of syllable number of 10. We hereafter called T1 as pauses detected by syntactic blocks having at least 5 syllables. The precision of this T1 prediction might be improved by further investigation. Besides, the recall of this prediction was rather low, i.e. 42.1%. We hence did more analyses on syntactic blocks having from 2 to 4 syllables for pause detection due to multiple ambiguous cases. Pauses predicted by syntactic blocks having from 2 to 4 syllables were called T2.

We carried out more study on some other predictors than syllable numbers of syntactic blocks, e.g. the level of syntactic blocks (first, second, etc.), number of syllables in previous/next syntactic blocks, position of syntactic blocks in sentence. Nevertheless, we could not discover a systematic relationship with pause presences. The two predictors, syntactic link and Part-Of-Speech (POS) at word level, were finally discovered for an improvement.

4.7.1 Proposal of syntactic link

The syntactic link of a word was a syntax tree-based relationship with the previous word. Four special values for this predictor "l1", "l2", "h1", "h2" show that if the current word was lower (l) or higher (h) one (1) or two (2) levels in the same branch. Some examples of syntactic link can be found in Figure 4.20. The syntactic link of the word B, illustrated in Figure 4.20a, was "h1" since the parent of the previous word A was a sibling of B (i.e. B was an uncle of A). The syntactic link of the word C was "l1" because the parent of C was a sibling of the previous word B (i.e. C was a nephew of B, or B was an uncle of C). In Figure 4.20b, the word F was a sibling of the grandfather of the previous word E hence the syntactic link of F was "h2". In Figure 4.20c, the syntactic link of the word G was "l2" since the grandfather of G was a sibling of the previous word F.

$$Distance(W_i, W_{i-1}) = D(W_i, W_{i-1}) = \frac{T_i + T_{i-1}}{2} \tag{4.11}$$

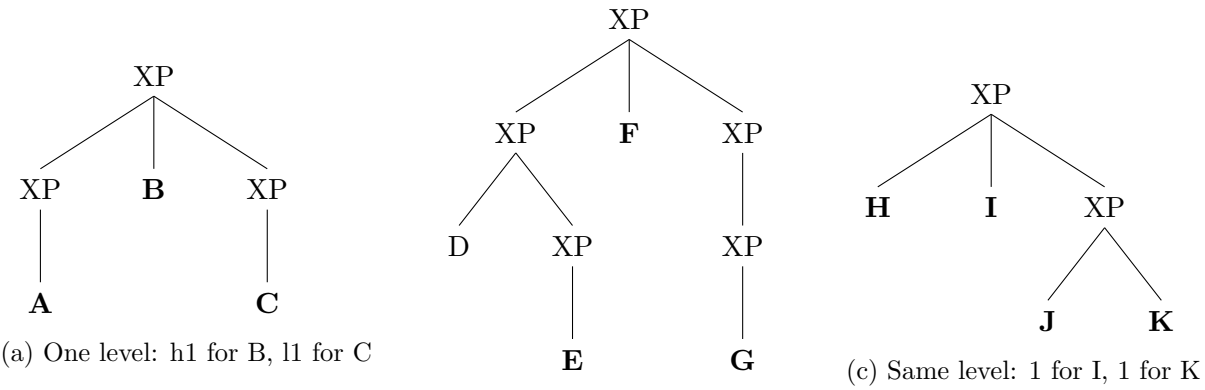
$$L(W_i) = L(W_i, W_{i-1}) = \begin{cases} \textbf{l1/l2} & \text{if } W_i \text{ was 1 or 2 level lower than } W_{i-1} \text{ (same branch)} \\ \textbf{h1/h2} & \text{if } W_{i-1} \text{ was 1 or 2 level higher than } W_{i-1} \text{ (same branch)} \\ \textbf{1} & \text{if } D(W_i, W_{i-1}) = 1 \text{ (siblings)} \\ \textbf{2} & \text{if } 1 < D(W_i, W_{i-1}) \leq 2 \\ \textbf{3} & \text{if } 2 < D(W_i, W_{i-1}) \leq 3 \\ \textbf{4} & \text{if } D(W_i, W_{i-1}) > 3 \end{cases} \tag{4.12}$$

$$NextL(W_i) = L(W_{i+1}) = L(W_{i+1}, W_i) \tag{4.13}$$

where

- W_i: The ith word in sentence
- W_{i-1}: The (i-1)th word in sentence
- T_i: Number of branch transitions from W_i to the lowest common ancestor of W_i and W_{i-1}
- T_{i-1}: Number of branch transitions from W_{i-1} to the lowest common ancestor of W_i and W_{i-1}.
- L: Syntactic link; NextL: Next syntactic link

In other cases, the distance between two nodes in a syntax tree was used to determine this relationship. The distance $D(W_i, W_{i-1})$ between two words W_i and W_{i-1} was calculated as a half of the total number of transitions from W_i and W_{i-1} to the lowest common ancestor, shown in Formula 6.1. If the distance was "1", two nodes were siblings (e.g. Figure 4.20c). The syntax link was the ceiling of this distance excluding the distance of over 3. For all distances over "3", one value "4" was assigned for syntactic link.



(a) One level: h1 for B, l1 for C

(b) Two levels: h2 for F, l2 for G

(c) Same level: 1 for I, 1 for K

Figure 4.20 – Example of syntactic links in syntax trees.

To summarize, the syntactic link $L(W_i)$ of a word W_i is defined in Formula 4.12. It was

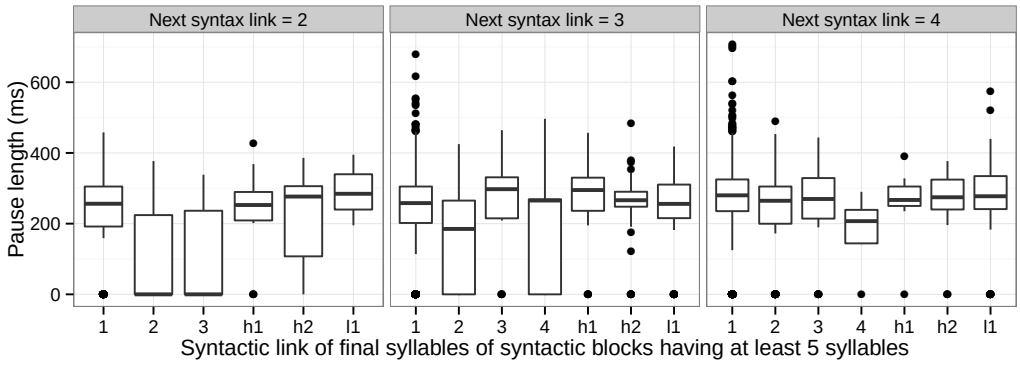
the syntactic link $L(W_i, W_{i-1})$ between the current word W_i and the previous word W_{i-1}. The next syntactic link $NextL(W_i)$ of a word W_i was the syntactic link between the next word and the current word $L(W_{i+1}, W_i)$, illustrated in Formula 4.13.

4.7.2 Rule-based model

Improvement with syntactic link. Based on the proposal of the new predictor "syntactic link", we assumed that the syntactic link between a word A and the next word B (that is the next syntactic link of the word A) gave a good information for their juncture. If the next word B and the current word A were loosely linked enough, there might be a pause between them.

To explore and confirm the above assumption, we plotted the distributions of pause length after the last syllables of syntactic blocks having at least 2 syllables by this predictor (see Figure B.10 in Appendix B). We found that few pauses appeared if the next syntactic link of the last syllable was not "2, 3, 4", the loosest values. As the assumption, most pauses were found to appear if the next syntactic link of the last syllables was "4", the loosest value. For less looser next syntactic links, i.e. "2" and "3", there were some ambiguous that need to be clarified.

(a) T1



(b) T2

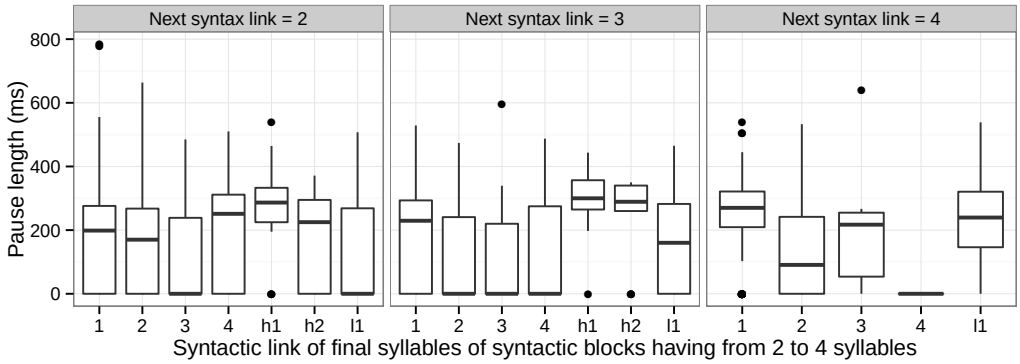


Figure 4.21 – Distributions of pause length after the last syllables of syntactic blocks having (a) at least 5 syllables; (b) from 2 to 4 syllables. The x-axis shows syntactic links of these last syllables, factored by their next syntactic links.

Similarly, the syntactic link of the current words showed its juncture with the previous word. If the current word and the previous word were tightly linked enough, there might be a pause after the current word. Figure 4.21 showed the distributions of pause length after

the last syllables of syntactic blocks having (a) at least 5 syllables (T1); and (b) from 2 to 4 syllables (T2). Distributions were displayed for syntax link and factored by the next syntactic link (from "2" to "4") of these last syllables.

After last syllables of syntactic blocks having at least 5 syllables (T1), few pauses appear when syntactic links were not tight enough, i.e. "2" or "3" (there was no observation for "4"). Some ambiguous cases of pause appearance were "2" and "4" of syntactic links when the next syntactic link was "3". For T2 pause prediction (syntactic blocks having from 2 to 4 syllables), pauses mostly appeared when syntactic links were tight enough, i.e. h1, h2, l1, 1 due to next syntactic links. Removing all cases with few pauses or ambiguous cases, the precision raised from 82.2% to 84.9%, recall increased 1.0% from 42.1% to 43.1% thus F-score enhanced 1.4%, from 55.7% to 57.1%. Detail of this fine-tuning and its results are presented in Table 4.13.

Improvement with syntactic link and POS. For further improvement, we did find that POSs of the last word (current POS) and that of the next word (next POS) of syntactic blocks could be used to predict pauses with ambiguity. By a preliminary analysis, we supposed that next POSs provided a better clue than current POSs to clarify those cases. We hence plotted the distributions of pause length of syntactic blocks having at least 2 syllables factored by next POSs (see Figure B.9 in Appendix B: (a) at least 5 syllables – T1, and (b) from 2 to 4 syllables – T2). The next POS was a really effective predictor when nearly two third of whose distributions clearly separated from zero points (no pause). Few pauses appeared if the next POS was one of those values "A", "E", "Nu", "R" and "V".

Some ambiguous cases, i.e. next POSs of "CC" in T1, or "L,M,T" in T2, could be elucidated by combining next POSs and current POSs. Distributions of pause lengths for those cases were discovered in Figure B.11, Appendix B. Ambiguous cases or cases with few pauses were removed.

Prediction rules using syntactic link and POS. With above analyses, prediction rules for pause appearance were constructed using the two predictors: syntactic link and POS.

Table 4.12 – Improvement of pause prediction with rules using syntactic link and POS predictors

Syllable # of syntactic blocks	Next-Current syntactic link	Next-Current POS	Correct predicted pauses#	Predicted pauses #	Precision
>=5 (T1)	!(2-2,2-3,3-2,3-4)	-	3,723	4,387	84.9%
2-4 (T2)	2-h1,3-h1,3-h2 or 4-1,4-3,4-l1	-	154	182	84.6%
>=5 (T1)	2-2,2-3 or 3-2,3-4 or 3-2,3-4	!(A-,E-,R-) and !(V-,Nu-) and !(CC-M,CC-V)	27	36	75.0%
2-4 (T2)	2-1,2-2,2-h2 or 3-1,3-l1,4-2	!(A-,E-,R-) and !(V-,Nu-) and !(L-V,L-A,M-R)	937	1,103	85.0%

Table 4.12 summarizes the improvement process with prediction rules using the three predictors: syntactic block, syntactic link and POS. The first two lines of this table present the improvement results for cases needing only syntactic links. The precisions after this improvement were nearly 85% for both types of pauses: T1–predicted by blocks having at

least 5 syllables, and T2–predicted by blocks having from 2 to 4 syllables. Prediction rules were extracted from the distributions of pause length after the last syllables of syntactic blocks.

The last two lines of Table 4.12 shows the improvements using "POS" for other ambiguous cases using the "Syntactic link" predictor. The performance of prediction rules for the T1 pauses was 75%, while that for the T2 ones was 85%. There were 1,118 additional T1 and T2 pauses (30% increase) predicted by using both syntactic link and POS predictors.

Table 4.13 shows the final results of improvement process. Along with the main predictor "syntactic block", the adoption of the "syntactic link" predictor helped the prediction performance increase about nearly 3% precision, and 1% recall. However, the recall of the improved model using both additional syntactic links and POSs was considerable improved, increased nearly 11% from 42.1% to 53.8%. The precision raised about 2.6% hence the F-score raises 10.1%, from 55.7% to to 65.8%.

Table 4.13 – Performance of three rule-based models for pause prediction using syntactic block, syntactic link, POS. T1 pau: pauses predicted by blocks having at least 5 syllables; T2 pau: pauses predicted by blocks having from 2 to 4 syllables

Model		Correct predicted pauses #	Predicted pauses #	Precision	Recall	F-score
Simple model with only syntactic blocks	T1 pau	3,790	4,609	82.2%	42.1%	55.7%
	T2 pau	0	0	-	-	-
	Both	**3,790**	**4,609**	**82.2%**	**42.1%**	**55.7%**
Syntactic-block and -link model	T1 pau	3,723	4,387	84.9%	41.3%	55.6%
	T2 pau	154	182	84.6%	1.7%	3.4%
	Both	**3,877**	**4,569**	**84.9%**	**43.1%**	**57.1%**
Syntactic-block and -link and POS model	T1 pau	3,761	4,438	84.8%	41.6%	55.9%
	T2 pau	1,091	1,285	84.9%	12.1%	21.2%
	Both	**4,841**	**5,708**	**84.8%**	**53.8%**	**65.8%**

4.7.3 Predictive model with J48

From above analyses, we assumed that the three important predictors for pause appearance were: (i) syllable blocks, (ii) syntactic links, and (iii) POS. Syntactic blocks played the most important role in the prediction. The rule-based model presented in the previous subsection was manually built for exploring and illustrating the importance of features. However, with such a dataset (i.e. VDTO-Analysis) with the three important predictors for each syllable and the actual subsequent pauses, we could apply a predictive technique to build an elaborate model (i.e. statistical classifier). This model can be automatically built for any speaker or any dialect by machine learning techniques.

In this work, J48, the Java implementation of the C4.5 algorithm, in the WEKA data mining tool[3] was adopted for experimenting different predictors due to its simplicity and effectiveness. The C4.5 algorithm made a number of improvements to the ID3 (Iterative Dichotomiser 3) algorithm, an algorithm invented by Quinlan (1986) used to generate a deci-

3. a collection of machine learning algorithms for data mining tasks: http://www.cs.waikato.ac.nz/ml/weka/

sion tree from a dataset. A decision tree is a predictive machine-learning model that decides the target value (dependent variable) of a new sample based on various attribute values of the available data. The internal nodes of a decision tree denote the different attributes, the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable. The attribute that is to be predicted is known as the dependent variable, since its value depends upon, or is decided by, the values of all the other attributes. The other attributes, which help in predicting the value of the dependent variable, are known as the independent variables or predictors in the dataset.

Using the WEKA tool, the extracted dataset including all syllables from the VDTO-Analysis and VDTO-Testing corpora with their features, such as POS, next POS, syntactic block size, the position in syntactic block, syntactic link, etc had to converted to a suitable format for WEKA (i.e. arff – attribute-relation file format). The experiment then could be easily performed with different predictors. J48 actually built a decision tree from predictors for a classification "Yes" or "No" for the target value "Has pause". Table 4.14 shows the performance of different models using J48 or rule-based approach in the previous subsection. With models using J48, the performance was actually of the Class "Yes" (i.e. pause appearance) in the results.

Table 4.14 – Performance of pause predictive models with J48 using different contextual features; and rule-based model

Model type	**Features**	**Test set**	**Precision**	**Recall**	**F-score**
Predictive with J48	Syntactic block	10-fold cross validation	83.4%	71.1%	76.8%
	Syntactic link		65.4%	43.7%	52.6%
	POS		73.4%	31.0%	43.6%
	Syntactic-block+link		83.4%	76.8%	80.0%
	Syntactic block+POS		87.2%	71.4%	78.6%
	Syntactic link+POS		70.6%	58.7%	61.4%
	Syntactic-block+link+POS		**89.0%**	**74.6%**	**81.2%**
	Syntactic-block+link+POS	**VDTO-Testing**	**87.6%**	**75.9%**	**81.4%**
Rule-based	**Syntactic-block+link+POS**		**84.9%**	**55.8%**	**67.4%**

We can observe performances of models with different predictors using J48 in 10-fold cross validation in Table 4.14. The "Syntactic block" was the most important predictor in both Precision (83.4%) and Recall (71.1%). The Precision of the model using "POS" alone (73.4%) was higher than that of the one using "Syntactic link" alone (64.4%). However, "Syntactic link" (Recall=43.7%) could predict more pauses than the "POS" predictor (Recall=31.0%). Using only "POS", the F-score of the model was only 43.6%, 9% lower than that using only "Syntactic link". This provided us the fact that "POS" was not a good feature for predicting pause appearance as other languages/other works (Keri et al., 2007) (Sarkar and Sreenivasa Rao, 2015). The combination of two out of the three predictors gave better results in Precision and/or Recall. The model with "Syntactic block" and "Syntactic link" had the same Precision to that with only "Syntactic block", but Recall increased nearly 6%. Whereas, the model with "Syntactic block" and "POS" almost had no progress on Recall, but about 4% higher on Precision. We assumed that "Syntactic link" helped us improve the Recall while "POS" gave effective information to increase the Precision. The complete model including the three predictors had the best results with Precision=89.0%, Recall=74.6%, hence F-score=81.2%.

The last two lines in Table 4.14 illustrates the performance of the "Predictive with J48" and "Rule-based" models with the three predictors but using a separate test set (VDTO-Testing). The Precision of the predictive one was a bit higher than the rule-based one (i.e. nearly 3%). However, the recall of the predictive model considerably improved, was about 14% higher than the rule-based one.

4.8 Conclusion

In this chapter, some analyses and statistical treatments using hierarchical syntactic information were performed to find out predictors for two levels of prosodic phrasing for Vietnamese TTS: (i) pause appearance, one of the most prominent and frequent levels; and (ii) final lengthening.

In a preliminary study, syntactic rules between constituent or dependent elements were proposed to predict three break levels after an iterative refinement process. Some statistical treatments of pause length and final lengthening at predicted boundaries were done for refining rules in a small corpus VNSP (630 sentences with manual annotations). These syntactic rules could work well, i.e. P=91.2% and F-score=69.7%, in the manual environment but only constituent syntactic rules gave an acceptable results (i.e. P=84.2%, F-score=39.9%) in the automatic environment. Furthermore, with an automatic syntax parser, predicted break levels did not differ significantly leading to only one break level, i.e. pause appearance.

Another approach, which was finally implemented to the final version of VTED, was proposed to use syntactic blocks for predicting final lengthening and pause appearance. Syntactic blocks were proposed as syntactic phrases whose sizes were bounded with a specific number of syllables. The analysis corpus was the VDTO corpus including 5,338 utterances in about 7.7 hours. Audio files in this corpus were automatically segmented at phoneme-level by EHMM labeler. Phonemes were then grouped to syllables and perceived pauses in a different tier. Text files were automatically parsed to syntax trees by the VTParser, the adopted Vietnamese syntactic parser using shift-reduce parsing with averaged perceptron.

A normalized syllable duration (ZScore) pattern of syntactic blocks was found to be similar to that of breath groups (containing syllables between two consecutive perceived pauses): (i) slightly shortening at the first syllable, (ii) shortening at the penultimate one, and (ii) strong degree of lengthening at the last one. Final lengthening even still existed but with a lower degree in the last syllables of syntactic blocks excluding last syllables of higher level ancestors and breath groups. As a result, two levels of prosodic phrasing using durational clues alone were identified: (i) pause appearance using syntactic blocks with a maximum of 10 syllables and (ii) final lengthening using syntactic blocks with a maximum of 6 syllables.

Improvements were done by some strategies of grouping single syntactic blocks for final lengthening. The pause appearance prediction was improved by combining the syntactic blocks with two additional predictors: (i) syntactic link, a syntactic-tree-based relationship/distance between two grammatical words; and (ii) POS. Some rules were finally constructed for predicting pause appearance.

The performance of the rule-based model with the three predictors was good, i.e. P=84.8% and F-score=65.8% at the analysis phase; P=84.9% and F-score=67.4% at the testing phase. A predictive model was experimented in a 10-fold cross validation with these three predictors using J48 (i.e. the Java implementation of the C4.5 algorithm, a decision tree approach for the classification problem). The "Syntactic block" was the most important predictor since the model with only this predictor had the best Precision (83.4%) and Recall (71.1%), compared to models with only POS (F-score=43.6%) or syntactic link (F-score=52.6%) alone. The

"Syntactic link" predictor helped the model improve the Recall (6% improved) while "POS" gave effective information to increase the Precision (4% improved). The complete model including the three predictors had the best results with Precision=89.0%, Recall=74.6%, hence F-score=81.2%. Using a separate test set (VDTO-Testing), the performances of the two models were slightly different. The Precision of the predictive model was a bit higher than the rule-based one (i.e. nearly 3%). However, the recall of the predictive model considerably improved, was about 14% higher than the rule-based one. This model could be automatically built for any speaker or any dialect by machine learning techniques, and hence was chosen for applying to the final version of VTED.

Chapter 5

VTED, a Vietnamese HMM-based TTS system

Contents

5.1 Introduction

Previous research on HMM-based Vietnamese TTS (Vu et al., 2009)(Kui et al., 2011)(Dinh et al., 2013)(Phan et al., 2013b, 2014) all adopted HTS framework for experiment. They presented only the core architecture from HTS (Zen et al., 2007), not the whole architecture of a Text-To-Speech system for Vietnamese. In this chapter, we present our work on designing a complete architecture and implementing VTED, an HMM-based TTS system for Vietnamese.

In this work, tonophones were used as a speech unit for the training and synthesis process. Lexical tones were also considered in designing contextual features. Section 5.2 gives an introduction of a typical HMM-based speech synthesis system. Based on the core architecture of HTS that only gives details of the training and synthesis parts, an entire architecture is presented in Section 5.3. In this architecture, a natural language processing (NLP) part is described in detail. The design of Vietnamese contextual features – crucial aspects for high-quality synthetic voice – is covered in Section 5.4. The development platform with main implementation configurations of VTED is presented in Section 5.5. The implementation of modules in the NLP part is explained in Section 5.6. Section 5.7 introduces different voices trained with VTED and different feature sets, which were used in perceptual evaluations.

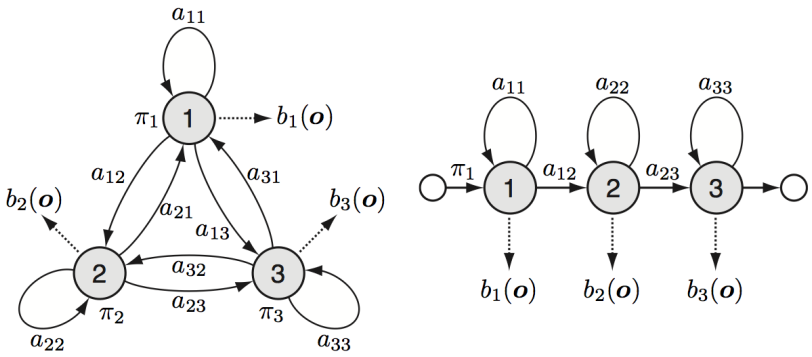
5.2 Typical HMM-based speech synthesis

As presented, in Figure 1.3 - Chapter 1, the core architecture of a typical HMM-based speech synthesis system, which was chosen for building a high-quality TTS system for the Vietnamese language, include two parts: training and synthesis parts (Yoshimura, 2002). In the training phase, first, spectral parameters and excitation parameters are extracted from speech database. The extracted parameters are modeled by a set of multi-stream context-dependent HMMs. In the synthesis phase, a context-dependent label sequence is first obtained from a given word sequence (i.e. input text) by text analysis. An utterance HMM is constructed by concatenating context dependent HMMs according to the context-dependent label sequence. Second, spectral and excitation parameters are generated from the sentence HMM by the speech parameter generation algorithm. Finally, a speech waveform is synthesized from the generated spectral and excitation parameters using a speech synthesis filter.

In this section, an overview of the Hidden Markov Model (HMM) was first introduced. Three main processes in this system are then described. They are (i) contextual features – important factors using in both training and synthesis part, (ii) parameter modeling – i.e. converting speech signals into a suitable representation, (iii) parameter generation – i.e. generating the corresponding speech waveform from the speech parameters, (iv) vocoder – for alleviating the "buzzy-sound" problem of the synthetic speech.

5.2.1 Hidden Markov Model

The hidden Markov model (HMM) is one of statistical time series models widely used in various fields. A hidden Markov model (HMM) is a finite state machine, in which the system being modeled is assumed to be a Markov process with unobserved (hence hidden) states. In an HMM, the state is not directly visible, but the output, dependent on the state, is visible. Each state has a probability distribution over the possible output observation. Therefore, the sequence of discrete time observations generated by an HMM gives some information about the hidden state sequence, the parameters of the model may be known.



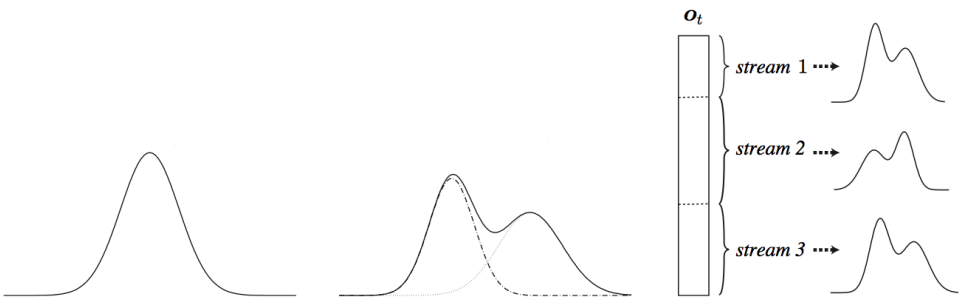
(a) A 3-state ergodic model (b) A 3-state left-to-right model

Figure 5.1 – Examples of HMM structure (Masuko, 2002).

When the HMM changes states at Markov process according to a state transition probability at each time unit (i.e., frame), observational data o_t at time t is generated in accordance with an output probability distribution of the current state. An N-state HMM is defined by the state transition probability $A = \{a_{ij}\}_{i,j=1}^{N}$, the output probability distribution $B = \{b_i(o)\}_{i=1}^{N}$, and initial state probability $\Pi = \{\pi_i\}_{i=1}^{N}$. Due to the simplicity, the parameter set of the HMM are denoted as shown in Equation 5.1 (Masuko, 2002).

$$\lambda = (A, B, \Pi) \tag{5.1}$$

Figure 5.1 illustrates two examples of the HMM structure: (a) a 3-state ergodic model, in which each state of the model can be reached from every other state of the model in a single transition, and (b) a 3-state left-to-right model, in which the state index simply increase or stay depending on the time increment. In speech processing, the left-to-right models are often used as speech units to model speech parameter sequences since they can appropriately model signals whose properties successively change.



(a) Gaussian PDF (b) Multi Mixture PDF (b) Multi Stream PDF

Figure 5.2 – Output distributions. PDF: Probability Density Function.

The output probability distribution $b_i(o)$ of the observational data o of state i can be discrete or continuous depending on the observations. For the continuous observational data, i.e. in the continuous distribution HMM (CD-HMM), the output in each state, a continuous value/vector with a Probability Density Function (PDF), is usually modeled by a mixture of multivariate Gaussian distributions (Figure 5.2b). When the observation vector o_t is divided into S stochastic-independent data streams (Figure 5.2c), $b_i(o)$ is formulated by product of Gaussian mixture densities.

5.2.2 Speech parameter modeling

Spectral and $F0$ modeling. In HTS, output vector of HMM composes of spectral part and excitation part. For example, the spectral part consists of melcepstral coefficient vector including the zeroth coefficients, their delta and delta-delta coefficients. On the other hand, the excitation part consists of log fundamental frequency (log $F0$), its delta and delta-delta coefficients.

For the spectral part, a continuous distribution HMM (CD-HMM) can be used for the vocal tract modeling in the same way as speech recognition systems. However, the $F0$ pattern is composed of continuous values in the "voiced" region and a discrete symbol in the "unvoiced" region. Therefore, it is difficult to apply either the discrete or the continuous HMMs to $F0$ pattern modeling. The work of Yoshimura (2002) proposed a kind of HMM for $F0$ pattern modeling, in which the state output probabilities are defined by multi-space probability distributions (MSD-HMMs). Figure 5.3 illustrates the spectral vector and $F0$ pattern, modeled by a continuous density HMM and multi-space probability distribution HMM, respectively.

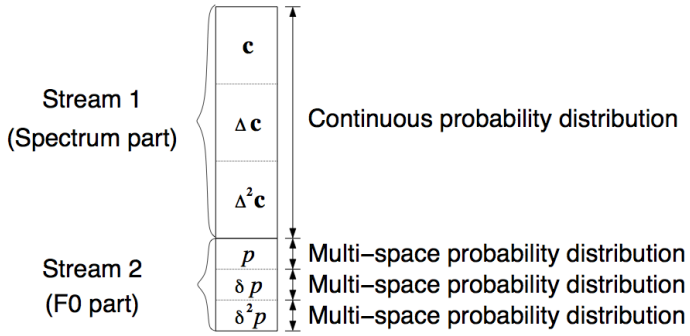


Figure 5.3 – Basic structure of a feature vector modeled by HMM (Yoshimura, 2002)

Duration modeling. HMMs have state duration densities to model the temporal structure of speech. State durations of each phoneme HMM are regarded as a multi-dimensional observation, and the set of state durations of each phoneme HMM is modeled by a multi-dimensional Gaussian distribution. Dimension of state duration densities is equal to number of state of HMMs, and n^{th} dimension of state duration densities is corresponding to n^{th} state of HMMs.

As a result, HTS models not only spectral parameter but also $F0$ and duration in a unified framework of HMM, as illustrated in Figure 5.4. In the synthesis part, an arbitrarily given text to be synthesized is converted to a context-based label sequence. Then, according to the label sequence, a sentence HMM is constructed by concatenating context dependent HMMs. State durations of the sentence HMM are determined so as to maximize the likelihood of the state duration densities. According to the obtained state durations, a sequence of mel-cepstral coefficients and F0 values including voiced/unvoiced decisions is generated from the sentence HMM by using the speech parameter generation algorithm. Finally, speech is synthesized directly from the generated mel-cepstral coefficients and F0 values by the MLSA (Mel Log Spectrum Approximation) filter (Yoshimura, 2002).

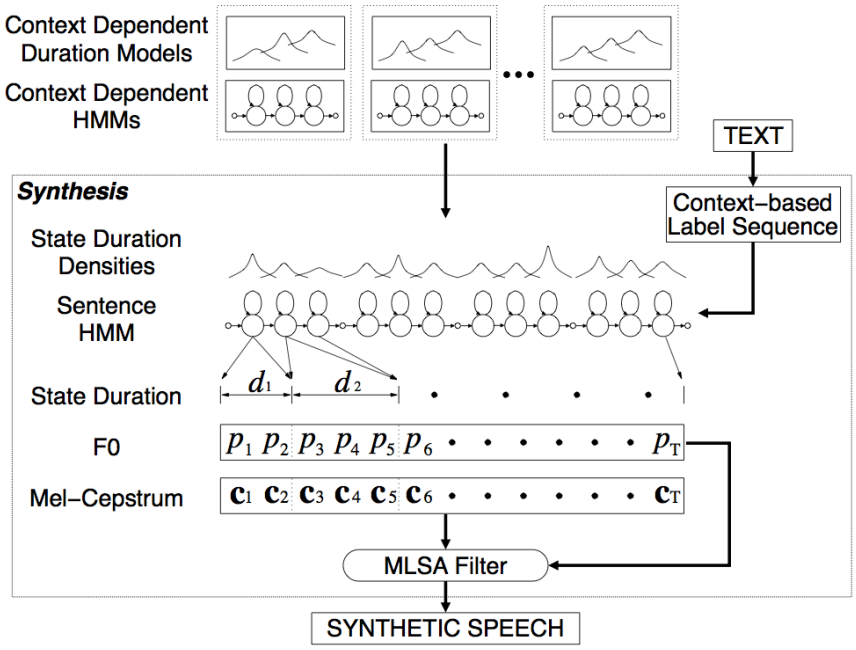


Figure 5.4 – Unified framework of HMM (Yoshimura, 2002).

5.2.3 Contextual features

The main difference of synthesis from recognition using the HMM-based approach is that linguistic and prosodic contexts are taken into account in addition to phonetic ones. There are many contextual features (e.g. phone identity features, stress-related features, locational features) that affect spectrum, F0 and duration. For example, the contexts used in the HTS English recipes in the work of Tokuda et al. (2002) were:

- Phoneme
  - current phoneme
  - preceding and succeeding two phonemes
  - position of current phoneme within current syllable
- Syllable
  - numbers of phonemes within preceding, current, and succeeding syllables
  - stress and accent of preceding, current, and succeeding syllables
  - positions of current syllable within current word and phrase
  - numbers of preceding and succeeding stressed syllables within current phrase
  - numbers of preceding and succeeding accented syllables within current phrase
  - number of syllables from previous stressed syllable
  - number of syllables to next stressed syllable
  - number of syllables from previous accented syllable
  - number of syllables to next accented syllable
  - vowel identity within current syllable

- Word
  - guess at part of speech of preceding, current, and succeeding words
  - numbers of syllables within preceding, current, and succeeding words
  - position of current word within current phrase
  - numbers of preceding and succeeding content words within current phrase
  - number of words from previous content word
  - number of words to next content word
- Phrase
  - numbers of syllables within preceding, current, and succeeding phrases
  - position of current phrase in major phrases
  - ToBI endtone of current phrase syllable
- Utterance
  - numbers of syllables, words, and phrases in utterance

Contextual features make HMM-based speech synthesis easily supporting multiple languages since these features to be used depend on each language.

Decision-tree based context clustering. There are many contextual features (e.g., phone identity, stress-related, locational features) that affect spectrum, $F0$ pattern and duration. To capture these effects, we use context dependent HMMs. However, as contextual features increase, their combinations also increase exponentially. Therefore, model parameters cannot be estimated accurately with limited training data. Furthermore, it is impossible to prepare speech database that includes all combinations of contextual features. To overcome this problem, a decision-tree based context-clustering technique is applied to distributions for spectrum, $F0$ and state duration in the same manner as HMM-based speech recognition.

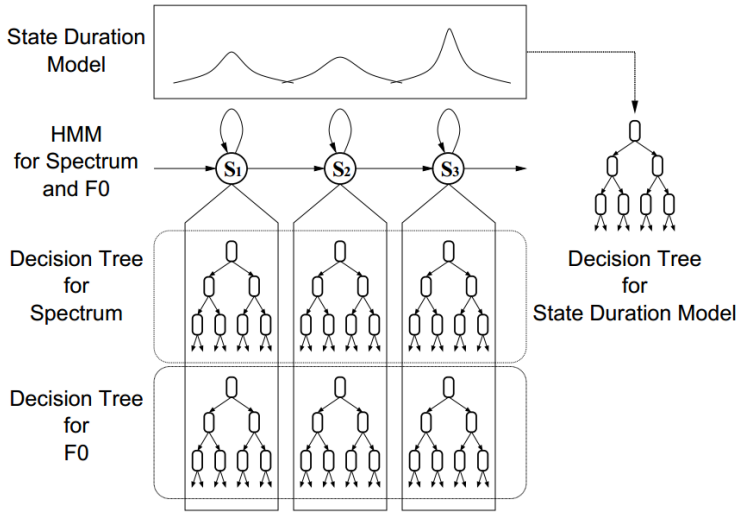


Figure 5.5 – Decision trees for context clustering (Yoshimura, 2002)

The decision-tree based context clustering algorithm have been extended for MSD-HMMs. Since each of spectrum, $F0$ and duration has its own influential contextual features, they are clustered independently (Figure 5.5). State durations of each HMM are modeled by a n-dimensional Gaussian, and context-dependent n-dimensional Gaussians are clustered by a decision tree. Note that spectrum part and $F0$ part of state output vector are modeled by multivariate Gaussian distributions and multi-space probability distributions, respectively.

5.2.4 Speech parameter generation

Before generating parameters, a state sequence is chosen using the duration model. "This determines how many frames will be generated from each state in the model. This would clearly be a poor fit to real speech where the variations in speech parameters are much smoother" (Zen et al., 2009). The speech parameter generation algorithm (e.g. the principal of maximum likelihood – the same criterion with which the model is usually trained) is used to generate a sequence of observations (i.e. vocoder/speech parameters: spectral, excitation parameters) from the utterance HMM. The speech parameter generation is actually the concatenation of the models corresponding to the full context label sequence, which itself has been predicted from text.

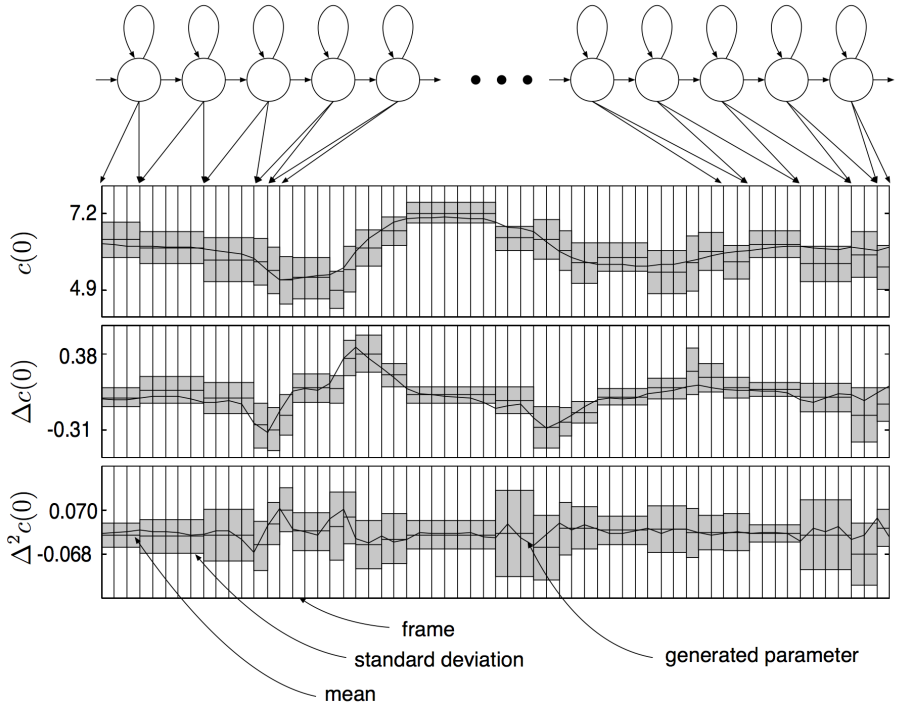


Figure 5.6 – Relation between probability density function and generated parameter for a Japanese phrase "unagi" (top: static, middle: delta, bottom: delta-delta) (Yoshimura, 2002). A smooth trajectory is generated from a discrete sequence of distributions, by taking the statistical properties of the delta and delta-delta coefficients into account.

To generate a realistic speech-parameter trajectory, the speech parameter generation algorithm introduces relationships between the static (e.g. cepstral coefficients) and dynamic features (e.g. delta and delta delta of cepstral coefficients) as constraints for the maximization problem (Zen et al., 2009). Figure 5.6 shows probability density functions and generated parameters for the zeroth mel-cepstral coefficients. The x-axis represents frame number. A gray

box and its middle line represent standard deviation and mean of probability density function, respectively, and a curved line is the generated zeroth mel-cepstral coefficients. It turns out that speech parameters are generated taking account of constraints of their probability density function and dynamic features (Yoshimura, 2002).

5.2.5 Waveform reconstruction with vocoder

Since the HMM-based synthetic speech has to be generated by a parametric model, "no matter how naturally the models generate parameters, the final quality is very much dependent on the parameter-to-speech technique used" (Taylor, 2009, p. 472). A vocoder plays an important role in the quality of an HMM-based speech synthesis system. The synthetic speech of a basic HMM-based speech synthesis system, which uses a mel-cepstral vocoder with simple periodic pulse-train or white-noise excitation, sounds "buzzy". Hence, a high-quality vocoder should be provided to alleviate this problem.

Traditional excitation model. A traditional excitation model included either a periodic impulse train or white noise shown in Figure 5.7. Synthesized speech has a typical quality of "vocoded speech" or "buzzy" sound with such a basic model.

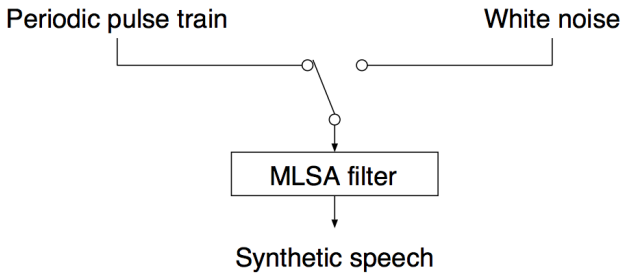


Figure 5.7 – Traditional excitation model.

Mixed excitation model. To overcome the above problem of the basic excitation model, a mixed excitation model was introduced in the work of Yoshimura (2002), Yoshimura et al. (2005) and postfilter for incorporating into the system. Excitation parameters used in mixed excitation are modeled by HMMs, and generated from HMMs by a parameter generation algorithm in the synthesis phase.

In the synthesis part of a HMM text-to-speech system, spectral and excitation parameter are output from the HMMs by applying speech applying speech parameter generation algorithms using the maximum likelihood criterion, with the synthesized speech produced by using the excitation source thus obtained to excite the synthesized filter constructed from the spectral parameters. To achieve a high quality synthesized sound, a high precision excitation model is necessary.

Figure 5.8 illustrates the mixed excitation model in the work of Yoshimura et al. (2005). Excitation parameters, i.e., F0, bandpass voicing strengths and Fourier magnitudes, are modeled by HMMs, and generated from trained HMMs in synthesis phase. This model involves the partitioning of the frequencies into bands, and the calculation of voicing intensity (bandpass voicing strength) separately for each frequency band. "If the bandpass voicing strength is above a certain threshold, that band is judged a voiced band, if it is below that threshold it is judged an unvoiced band. Pulse sequences are assigned to bands judged to be voiced while

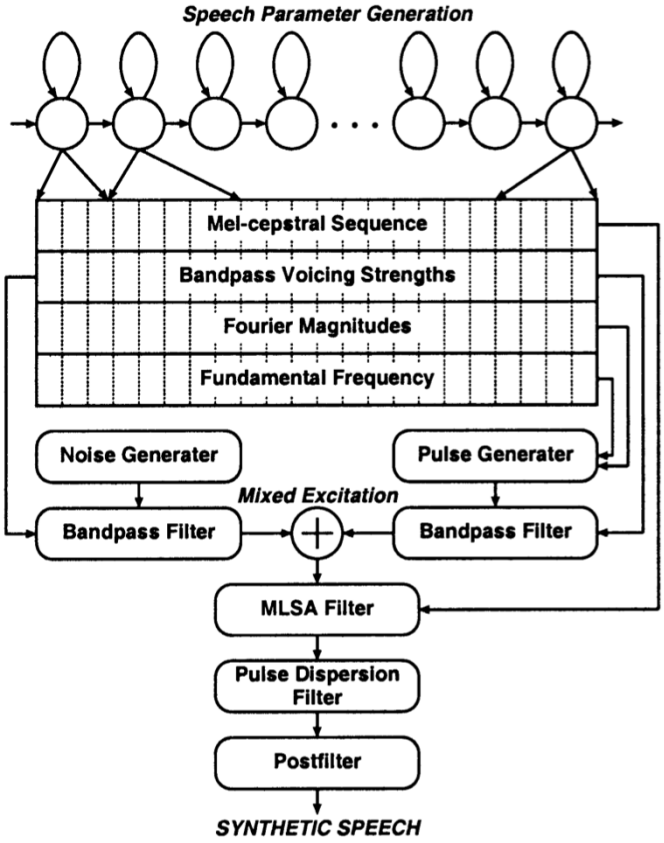


Figure 5.8 – Mixed excitation model (Yoshimura et al., 2005).

white noise is assigned to bands judged to be unvoiced. The mixed excitation is obtained by passing each of these through a bandpass filter and combining them together".

By exciting the MLSA filter, synthesized speech is generated from the mel-cepstral coefficients, directly. The obtained speech is filtered by a pulse dispersion filter, which can reduce some of the harsh quality of the synthesized speech. Finally, using a post-filter, the clarity of the synthesized voice can be improved by emphasizing formants.

5.3 Proposed architecture

The proposed architecture of an HMM-based TTS system for the Vietnamese language is illustrated in Figure 5.9. There were three parts in this architecture: (i) Natural language processing (NLP), (ii) Training, and (iii) Synthesis. From the input text, the NLP part extracted contextual features to provide for both Training and Synthesis phases. In the Training phase, these features were then aligned with speech unit labels and trained with speech parameters (i.e. spectral and excitation) to build context dependent HMMs. In the Synthesis phase, according to a label sequence with these factors, contextual features were used to produce a sequence of speech parameters. Finally, a synthetic speech was obtained using these speech parameters and a vocoder.

It appeared that the contextual features played a crucial role for the TTS system; hence they will be presented in detail in the next section. The following subsections describe each part of this architecture. The training and synthesis parts are similar to the core architecture

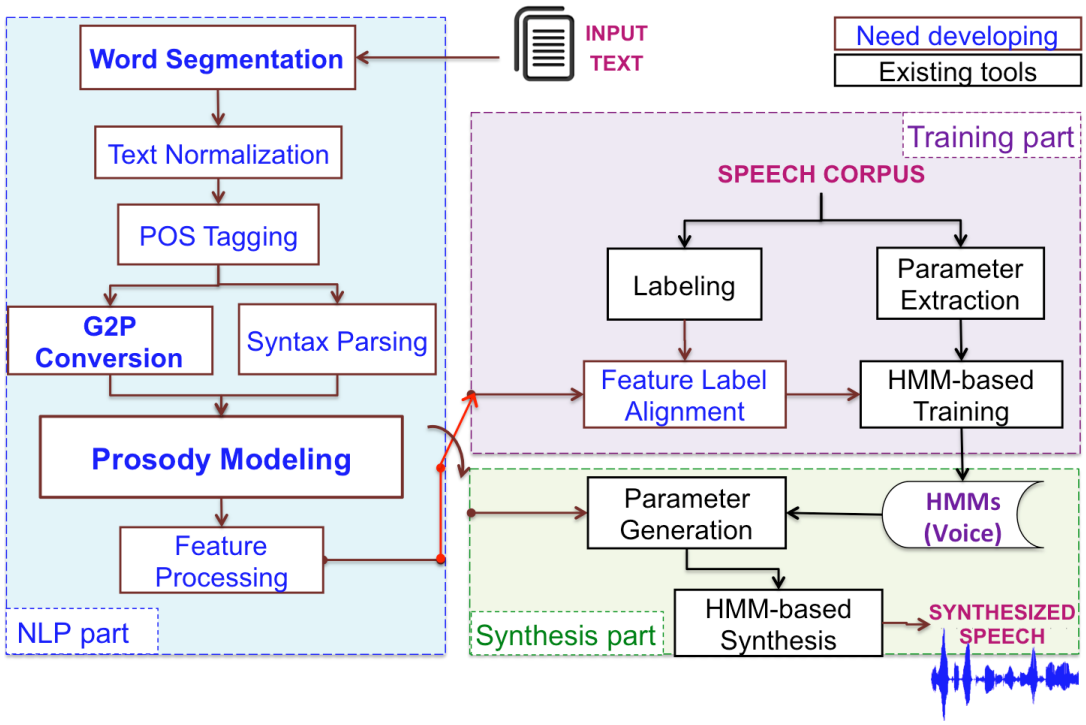


Figure 5.9 – Proposed architecture of the HMM-based TTS system for Vietnamese.

of HTS in the previous section.

5.3.1 Natural Language Processing (NLP) part

As illustrated in Figure 5.9, the NLP part accepted text as input for segmentation into normalized words, which were then tagged with Part-Of-Speech (POS). Next, the intermediate results were parsed to syntax trees and converted to phonemes for prosody modeling. The NLP part finally produced phonetic, linguistic and prosodic contextual features to both the Training and Synthesis phases. In our NLP part, there were seven modules.

The Word Segmentation module split the input text into sentences and then words. This module might simply replace blanks with word boundaries and cut off punctuation marks, parentheses and quotation marks at both ends of a word. However, with some languages having linguistic mechanism of syllabic scripts, like Vietnamese or Chinese, there are ambiguities in identification of what constitutes a word in an input text.

The Text Normalization module tried to convert all non-standard words (such as numbers, dates, abbreviations, currency) to pronounceable words. Like other languages, there exist a number of ambiguous cases where many items have more than one plausible pronunciation, and the correct one must be disambiguated by context.

The POS tagging module gave the part-of-speech (POS) for words from the previous module. This module might be simplified by giving the classification of functional/content words or provide a full POS tag set.

Next, each word was transcribed to tonophones, and tone was extracted at syllable-level by the G2P Conversion. Vietnamese is an isolating language, the most extreme case of an analytic language, in which the boundaries of syllables and morphemes are the same (each morpheme is a single syllable). Furthermore, syllables have a stable structure with a set of

quite-well-defined G2P rules. Although there exist exceptional or complex cases for that, it can be said that G2P is not a big problem for Vietnamese. These tagged sentences were then parsed into syntax trees by a syntax parser. The syntax trees could be used for the prosody modeling.

Information obtained from all previous tasks was processed to be contextual features corresponding the input word sequence by the Factor Processing module. These features included contextual features in phoneme, syllable, word, phrase, and utterance level with respect to the lexical tones.

Detail of each module will be described in the next sections.

5.3.2 Training part

In the Training part, the speech corpus was automatically segmented and labeled with the transcription of the input text by a labeler, such as EHMM or Sphinx [1]. These labels were force-aligned with the contextual features extracted from the NLP part. Speech parameters including spectral (e.g., mel-cepstral co-efficients and their dynamic features) and excitation parameters (e.g., log F0 and its dynamic features) were automatically extracted using existing tools, such as Snack [2], SPTK [3], WinPitch (Martin, 2000, 2004, 2005), etc.

The two inputs of the HMM-based Training module were: (i) Extracted speech parameters, and (ii) contextual features aligned with labels of the speech corpus. This process performed the maximum likelihood estimation by using the Expectation Maximization algorithm, and produced a trained voice of context-dependent HMMs and duration models.

5.3.3 Synthesis part

In the Synthesis part, contextual features were used to build a context-dependent label sequence. According to this label sequence, a sentence HMM was constructed by concatenating context dependent HMMs in such a way that its output probability for the HMM was maximized. Spectral and excitation parameters were generated from the sentence HMM by the parameter generation algorithm. Finally, synthesized speech was obtained using these speech parameters and a high-quality vocoder.

5.4 Vietnamese contextual features

5.4.1 Basic Vietnamese training feature set

As presented in Section 2.6 – Chapter 2, a tonophone can reflect the tonal context of the respective allophone. Hence, it was chosen as a speech unit for our HMM-based Vietnamese TTS system. There were totally 207 units in the tonophone set whose detail can be found in Section 2.6, Chapter 2.

Basic phonetic and prosodic features. Contextual features for Vietnamese were chosen at tonophone, syllable, word, phrase and utterance levels, based on Vietnamese phonetic and phonology in Chapter 2. We also referred to the work on English (Tokuda et al., 2002) and

1. a speaker-independent and continuous speech recognition system: original version: http://cmusphinx.sourceforge.net/, state-of-the-art version: https://github.com/cmusphinx/sphinx4
2. The Snack Sound Toolkit: http://www.speech.kth.se/snack/
3. Speech Signal Processing Toolkit: original version: http://sp-tk.sourceforge.net/, state-of-the-art version: https://github.com/r9y9/SPTK

on Vietnamese (Vu et al., 2009)(Kui et al., 2011) to build our own feature set. We had more features on phone positions in syllable structures, punctuations and some prosodic features (which are formatted in *italic*). There were some slightly differences on number of sub-levels and relative positions of each level.

As mentioned in Chapter 2, Vietnamese syllables have a stable and complete structure composed of four elements: initial, medial, nucleus and ending. Although only nucleus is mandatory, the appearance of other elements is also common. To capture the context of tonophones between elements inside a syllable as well as the transitions to previous/next syllables, we used the two preceding and two succeeding tonophones for phonemic context of the current tonophone.

About 84% Vietnamese words are compound words, which compose of at least two syllables. About 70% of compound words have two syllables, only 1% of compounds have more than four syllables. Therefore, we only used one preceding and one succeeding syllable or higher levels for its context.

As a result, following contextual features at five levels were taken into account in this work:

- Tonophone level
  - {Two preceding, current, two succeeding} tonophones
  - The tonophone is onset or coda
  - Number of tonophones {from the beginning, to the end} in the current syllable to the current tonophone.
  - *Break indices of the current tonophone.*
- Syllable level
  - Number of tonophones in the {preceding, current, succeeding} syllable
  - Number of syllables {from the beginning, to the end} in the current word to the current syllable.
  - *Lexical tone of {preceding, current, succeeding} syllable*
  - *Position type of the current syllable*
- Word level
  - Number of {tonophones, syllables} in the {preceding, current, succeeding} word
  - Number of words {from the beginning, to the end} in the current phrase to the current word
  - *{Previous, next} punctuation in the current sentence*
  - *Part-of-Speech (POS) tags of the {preceding, current, succeeding} word*
  - *Number of words from the {preceding, succeeding} punctuation in the current sentence*
- Prosodic phrase level
  - Number of {syllables, words} in the {preceding, current, succeeding} phrase
  - Number of words {from the beginning, to the end} of the current utterance
- Utterance level

– Number of {words, phrases} in the {preceding, current, succeeding} utterance
– Punctuation of the {preceding, current, succeeding} utterance

The lexical tone of a syllable could be one of 8 values represented for 8 tones: 1-4, 5a, 5b, 6a and 6b. Position type of a syllable could be "single" if there is only one syllable in the bearing word; "initial", "middle" or "last" corresponding to its position in a multi-syllable word. POS value of a word could be one of the list in Table 4.3, Section 4.3, Chapter 4. In our preliminary experiment, prosodic phrases were identified by punctuations in the middle of sentences, i.e. ", ; : ( ) " ' ".

At tonophone level, there were 6 values for the break levels, which were discovered by simple prosodic phrasing rules, illustrated in Table 5.1. The "utterance boundaries" (break indice 5) can be automatically identified by sentence punctuations (e.g. ". ? !") while the "within-word boundaries" (break indice 0) can be detected betwen two consecutive tonophones in one grammatical word. The "phrase-medial word boundaries" (break indice 1) were at the edges of two consecutive words. The intonation phrase boundaries (break indice 4) were identified as the same as the prosodic phrases, i.e. punctuations in the middle of sentences, i.e. ", ; : ( ) " ' ". There were no explicit rule for the immediate phrase boundaries but two break indices were kept for further process.

Table 5.1 – Prosodic phrasing rules

Break index	**Boundary name**	**Rule**
0	Within-word boundary	Between 2 consecutive tonophones in one word
1	Phrase-medial word boundary	Between 2 consecutive words
2, 3	Intermediate phrase boundary	N/A
4	Intonation phrase boundary	After a punctuation mark in the middle of the sentence
5	Utterance boundary	At end of sentence, not at end of paragraph

5.4.2 ToBI-related features

The ToBI (Tones and Break Indices) system (Silverman et al., 1992), which is intended as a standard for the prosodic transcription of American English, is also supposed to be compatible with current work in language processing, explicitly modeling two crucial prosodic cues i.e. intonation (boundary tones) and juncture (break indices). However, prosodic features predicted from ToBI model did not work well for Vietnamese, due to the constraint of the utterance-level intonation and lexical tones. The ToBI transcription model can be found in detail in Section C.1, Appendix C.

Several ToBI-related features might include phrasal tones, pitch accents and break index. The ToBI break index was already considered in the basic feature set for Vietnamese, as illustrated in Table 5.1. For intonation phrase boundaries, break index was triggered only by punctuations in the middle of sentences. ToBI Pitch accent tones were marked at every accented syllable (Beckman and Hirschberg, 1994). In more recent phonetic research, the terms "stress" and "accent" both came to be regarded as being physically manifested, "stress" being the default realization of a lexically stressed syllable outside focus, as determined by word phonology, "accent" its realization for contrast under narrow focus, superimposed on its default properties (Nguyen and Ingram, 2013). However, "Vietnamese is a contour tone

language that has no system of culminative word stress; nevertheless, it is widely accepted that there is stress in the sense of accentual prominence at the phrasal level" (Thompson, 1987). As a result, pitch accent tones were not adopted in our features, after a preliminary test. Only ToBI phrasal tones were assigned at every intermediate or intonation phrase (Beckman and Hirschberg, 1994). In our training feature set, phrasal boundary tones of the {preceding, current, succeeding} phrase were chosen at phrase level.

The study on Vietnamese intonation by (Do et al., 1998) discussed some works on the intonation of declarative and interrogative sentences. These works described in a qualitative way for these sentences mode. Declarative sentences were discussed with a falling intonation, F0 declination or "low speech", whereas interrogative sentences were said to be rising contour or "higher pitch". The significant difference between sentence modes would relate to average register, which would be situated in the middle of the range for declaratives and towards the periphery for other sentence modes. Other work (Le et al., 2011) confirmed that the F0 contour of the last syllable or the one of its second half tended to increase for questions.

Table 5.2 – ToBI boundary tone (i.e. intonation rules) for phrases

Boundary mode	**Description**	**Rule**
EX	End of exclamative sentence	L-H%
IN	End of interrogative sentence	H-H%
DE	End of declarative sentence	L-L%
EM	End of a phrase, terminated by a punctuation mark in the middle of the sentence	H-L%

Rules for ToBI boundary tones for phrases in Table 5.2 were built from the ones for American English (Jilka et al., 1999) for Vietnamese regarding above analyses on the intonation of Vietnamese interrogative sentences. Three sentence modes were then experimented in this work: declarative, exclamative and interrogative sentence (Table 5.2). In declarative mode, sentence-internal boundaries were labeled L-H% and sentence final boundaries were labeled L-L%. Interrogative sentences were transcribed with H-H% while exclamative sentences were labeled with L-H% pattern. Phrases terminated by a punctuation mark in the middle of the sentence had a H-L% pattern.

5.4.3 Prosodic phrasing features

As presented in Chapter 4, we proposed a prosodic phrasing model with two levels: (i) pause appearance using POS, syntactic link and syntactic blocks with a maximum of 10 syllables (Section 4.7), and (ii) final lengthening using syntactic blocks with a maximum of 6 syllables (Section 4.6). Several new prosodic training features for these two levels were proposed for Vietnamese HMM-based speech synthesis.

For final lengthening, new proposed prosodic features using syntactic blocks with a maximum of 6 syllables, shown in Table 5.3, were: number of syntactic blocks in the current sentence, position of the current syntactic block in the current sentence, number of syllables in the current syntactic block, position of the current syllable in the current prosodic block. The procedure for identifying these syntactic blocks can be found in Listing B.1, Appendix B.

Pause appearance was detected using the J48 decision tree of WEKA with the three predictors including POS, syntactic link, and syntactic blocks with a maximum of 10 syllables. Along with punctuations in the middle of sentences, predicted pauses were also used as

Table 5.3 – New training features for final lengthening from syntactic blocks

New prosodic features	**Value range**	**Identification rules**
Number of syntactic blocks in sentence	1..20	Number of syntactic blocks in the current sentence, limit to 20 if above
Position of syntactic block in sentence	1..20	Position of the current syntactic block in the current sentence, limit to 20 if above
Number of syllables in syntactic block	1..6	Total number of syllables in the current syntactic block
Position of syllable in syntactic block	1..4	- If the current syntactic block has 1 syllable: **1** - If the current syntactic block has at least 2 syllables: + if current syllable is the first: **1** + if current syllable is the last: **4** + if current syllable is the penultimate: **3** + if current syllable is the middle: **2**

the boundaries of prosodic phrases (i.e. intonation ones). With this prediction, contextual features at the prosodic phrase level could be extracted better.

5.5 Development platform and configurations

5.5.1 Mary TTS, a multilingual platform for TTS

There are several platforms that can be used to develop an HMM-based TTS system. The most common one is HMM-based Speech Synthesis System (HTS) [4], whose training part has been implemented as a modified version and released as a form of patch code of HTK [5]. HTS does not include any text analyzers but the Festival Speech Synthesis System (English, Spanish, etc.), MARY TTS (German, English, etc.), Flite+hts_engine (English), Open JTalk (Japanese), or other text analyzers can be used with HTS.

Open JTalk is a Japanese text-to-speech system. Festival is a free software multi-lingual speech synthesis workbench that runs on multiple-platforms offering black box text to speech, as well as an open architecture for research in speech synthesis. It designed as a component of large speech technology systems. The drawbacks of Festival are incomplete and not updated technical docs, big size and slow speed. Flite (festival-lite) is a small, fast run-time synthesis engine developed at CMU and primarily designed for small-embedded machines and/or large servers. Flite is designed as an alternative synthesis engine to Festival for voices built using the Festvox suite of voice building tools [6]. It currently supports English and Indic.

Mary TTS (Modular Architecture for Research on speech sYnthesis) is an open-source, multilingual TTS platform written in Java. It is maintained by the German Research Center for Artificial Intelligence (DFKI) [7]. Mary TTS comes with toolkits for quickly adding support for new languages and for building unit selection and HMM-based synthesis voices. MaryTTS now has a big development community in GitHub [8] with number of advantages such as complete and updated technical docs, clear architecture and source code (pure object-orientation with Java). The system, the result of years of development, is really huge and complex with hundreds of packages, thousands of classes (more than 260,000 lines of Java code), and other resources.

Due to the facilities and expendability, Mary TTS was ultimately chosen as a platform to build an HMM-based TTS system for Hanoi Vietnamese – VTED. This system supports the client/server speech synthesis via HTTP. The client sends input data to the server and receives processing results. The server is multi-threaded, which allows for more than one client or component to send requests to be processed. The client allows users to choose from different input types (e.g. plain text, the SSML [9], words, phonemes, etc.) and output types (e.g. audio, words, phonemes, ect.). The two far ends are plain text as input and audio as output, however it is possible to output intermediate processing steps as well.

5.5.2 Mary TTS workflow of adding a new language

The workflow supporting for adding a new language of Mary TTS [10] is illustrated in Figure C.1, Appendix C. At least basic NLP package need to be developed for the new language. The minimal NLP package must provide a module to convert from text to allophones (Phonemizer), which may use a pronunciation lexicon and letter-to-sound rules for unknown words. Furthermore, this NLP package also needs to provide a rudimentary POS tagger module.

4. http://hts.sp.nitech.ac.jp/
5. http://htk.eng.cam.ac.uk/
6. http://www.festvox.org/flite/
7. http://mary.dfki.de/
8. https://github.com/marytts/marytts
9. Speech Synthesis Markup Language: http://www.w3.org/TR/speech-synthesis11
10. https://github.com/marytts/marytts/wiki/New-Language-Support

Other modules such as Tokenizer or Prosody Modeling were generic implemented with basic functionality for all languages. The whole workflow can be performed by a voice builder. This tool supports functions for extracting clean text from Wikipedia dump using mysql database for the new language. The most frequent words can be also extracted for manually being transcribed. This tool provides a function to select and record sentences for the training corpus. Finally, an unit selection or HMM-based voice can be built for synthesis.

After the training process, a training voice of a corpus recorded by specific speaker was obtained. This voice can be used to synthesize by the *HMM-based synthesizer* module, which was ported to Java from the HTS (Schröder et al., 2008).

5.5.3 HMM-based voice training for VTED

This section gives an introduction of main modules for HMM-based voice building of Mary TTS. This platform provides a graphical user interface (GUI) for all steps whose technical tutorial can be found on the wiki page of MaryTTS GitHub [11], illustrated in Figure 5.10 (Würgler, 2011). Parameters and configurations of each step can be set using the GUI.

The main modules (by the execution order) in the voice building process of Mary TTS are described as follows.

The *Allophones Extractor* processes the text transcription of the speech signals to generate a MaryXML allophones file (initial allophones/tonophones). This component requires the MARY server running.

The *EHMM Labeler* is an external component called by MARY for automatic phonetic segmentation and labeling of the speech signals. In this tool, continuous models with one Gaussian per state, left-to-right models with no skip state and context-independent models trained with 13 MFCCs are used to get force-aligned labels (Schröder et al., 2008). We had described this step in the corpus design (cf. Chapter 3).

The *Phone Unit Label Computer* converts the label format from EHMM to MARY. *Transcription Aligner* makes sure that the label and the allophone files are aligned, thus no mismatch between the phone sequences exists (final allophones/tonophones).

The *Feature Selection* confirms a list of all contextual features to be considered in the next steps and saves to a feature file. This module automatically extracts linguistic or contextual feature vectors from the allophone files. For each phone it basically lists all kind of features, mostly phonetic (vowel height, consonant type, etc) and contextual features from the previous step.

The *Phone Unit Feature Computer* extracts context feature vectors from the text data. To run this component the MARY server should be running as well. The *Phone Label Feature Aligner* makes sure that the phone labels and the phone feature files are aligned, thus no mismatch between the phone sequence exists.

The *HMM Voice Data Preparation* sets up the environment to create a HMM voice and check if the required external programs, text and wav files are available and in the correct paths. It converts all wave files to raw by using the SoX tool [12]. In this step, the sample rate is an important setting for a correct training. ***For VTED, with the old corpus VNSP, the sample rate was set to 16,000 Hz while it was 48,000 Hz for the new corpus VDTS***.

The *HMM Voice Configure* configures some voice properties, e.g. its name, sampling rate, lower/upper frequency bound (depending on male or female voice), etc. Some default setting

11. https://github.com/marytts/marytts/wiki/HMMVoiceCreation
12. Sound eXchange: http://sox.sourceforge.net/

values are already fixed for the default voice "arctic slt". ***For VTED, the lower F0 bound was set to 75 Hz, the upper one 450 Hz****, after observing the signal to see some highest and lowest values of F0 (both speakers of the VNSP and VDTS corpora were female).* ***A tonophone was represented by the default 5-state left-to-right HMM structure of Mary TTS***. Table 5.4 illustrates basic HMM configurations of VTED for two corpora with different speakers and sample rates. The old corpus VNSP was recorded by a TV broadcaster speaker at 16 kHz and 16 bps, while the new one was recorded by a female non-professional speaker Thu-Trang from Hanoi at 48 kHz, 24 bps and eventually converted to 48 kHz, 16 bps. Some other different setting values shown in Table 5.4 for the two corpora were: (i) FFT length, (ii) Frame length, (iii) Frame shift, and (iv) Frequency warping. These values were calculated based on the sample rates of corpus.

Table 5.4 – Some HMM configuration values for VTED

Corpus	Speaker	Sample rate (Hz)	FFT length	Frame length	Frame shift	Frequency warping
VNSP (old)	Broadcaster	16,000	512	400	80	0.42
VDTS (new)	ThuTrang	48,000	2,048	1,200	240	0.55

The *HMM Voice Feature Selection* saves a list of features that are used to build the context tree for the HMM models. This could basically be the same list as above (i.e. Feature Selection module). However, a subset of features is usually chosen. ***For VTED, different feature sets presented in Section 5.7 were settled on different version of training voices***.

The *HMM Voice Make Data* basically calls external HTS scripts, which have been slightly adapted to the MARY system. This step uses the SPTK and Snack tools for extracting speech parameters including mel-generalized cepstral coefficients (mgc), log F0 (lf0), voicing strengths for mixed excitation (str) and Fourier magnitudes (mag) from the audio files. They are assembled to an acoustic parameter vector (mgc+lf0+str+mag).

The *HMM Voice Make Voice* uses a version of the speaker dependent training scripts provided by HTS that was adapted to the MaryTTS platform to train HMMs. These training scripts have been modified to use (i) context features predicted by the NLP part, (ii) global variance (iii) composed training data of mgc, log F0, voicing strengths and Fourier magnitudes (from the previous step) for generation of mixed excitation (Pammi et al., 2010).

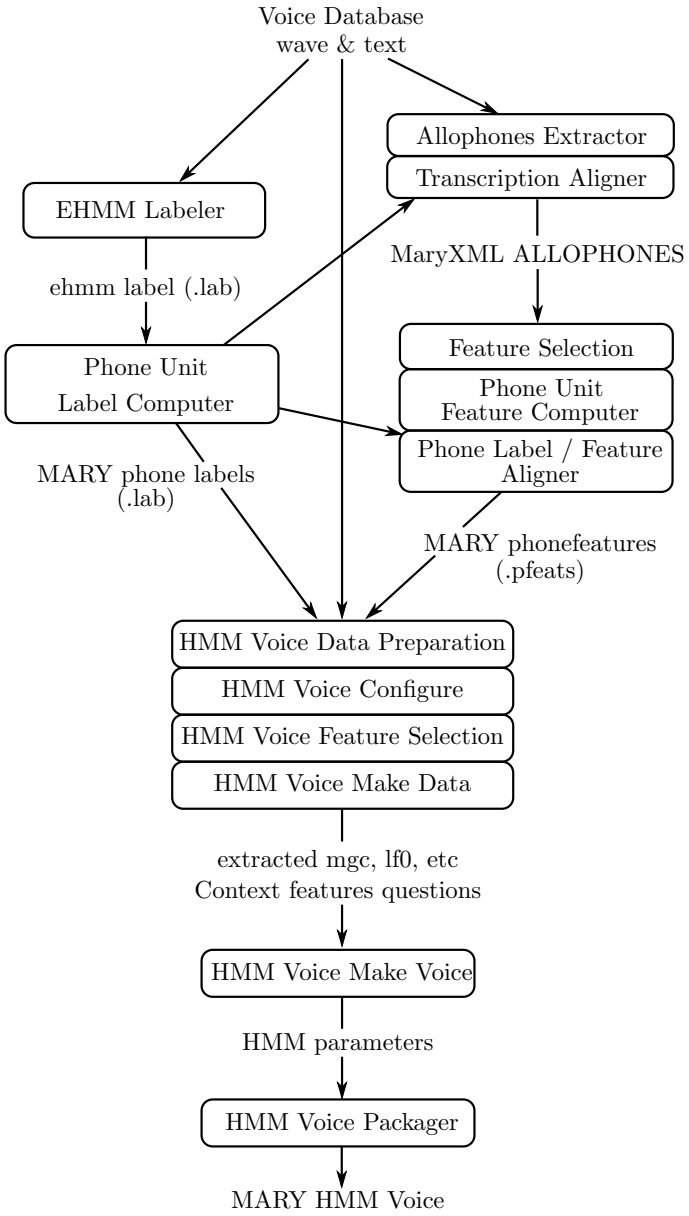


Figure 2.1: MARY HMM-based voice training.

Figure 5.10 – HMM-based voice training in Mary TTS (Würgler, 2011).

5.6 Vietnamese NLP for TTS

Seven modules in the NLP part were: (i) Word Segmentation, (ii) Text Normalization, (iii) POS Tagging, (iv) G2P and Tone Extraction, (v) Syntax Parsing, (vi) Prosody Modeling, and (vii) Feature Processing. The following subsections describe in detail these modules for the Vietnamese language.

5.6.1 Word segmentation

As aforesaid, in Vietnamese, each syllable usually has an independent meaning in isolation, and polysyllables can be analyzed as combinations of monosyllables (Doan, 1977). Hence, a syllable in Vietnamese is not only a phonetic unit but also a grammatical unit (Doan, 1999b). Although dictionaries contain a majority of compound words, monosyllabic words actually account for a wide majority of word occurrences. This is in contrast to synthetic languages, like most Western ones, where, although compound words exist, most words are composed of one or several morphemes assembled so as to form a single token (Le et al., 2010). In other words, the Vietnamese language creates words of complex meaning by combining syllables that most of the time also possess a meaning when considered individually. This linguistic mechanism makes Vietnamese close to that of syllabic scripts, like Chinese. That creates problems for all natural language processing tasks, complicating the identification of what constitutes a word in an input text (Le et al., 2008, p. 240).

Although Vietnamese is an alphabetic script, unlike occidental languages, "blanks are not only used to separate words, but they are also used to separate syllables that make up words". Moreover, many of Vietnamese syllables are not only words by themselves, but can also be part of multi-syllable words whose syllables are separated by blanks between them (Le et al., 2008). For instance, in the phrase "thế hệ những kiến trúc sư đầu tiên của Việt Nam" (the first generation of Vietnamese architects), there are 1 trisyllable word (i.e. "kiến trúc sư" - architect)), 3 disyllable words (i.e. "thế hệ" - generation, "đầu tiên" - first and "Việt Nam" - Vietnam) and two single syllable word (i.e. "những" - plural, "của" - of).

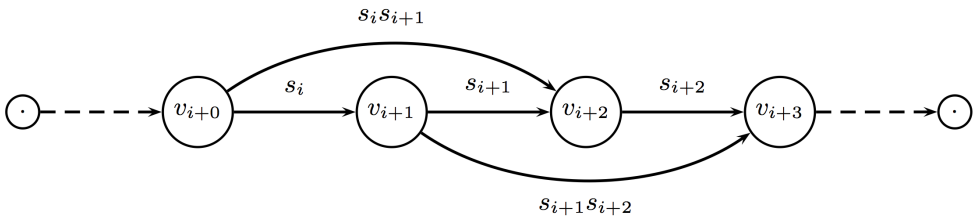


Figure 5.11 – Overlap ambiguity of Vietnamese word segmentation (graph representation) (Le et al., 2008).

In other words, there is no explicit word delimiter in Vietnamese, i.e. no specific marker distinguishes the spaces between actual words. For example, a simple sequence of three syllables "a b c" can constitute three words (a) (b) (c), two words (a b) (c), two words (a) (b c) or even a single one (a b c). This characteristic leads to many ambiguities in word segmentation, the "foremost basic processing task" that influences part-of-speech tagging and higher levels of natural language processing for Vietnamese text. As an example, a phrase of three syllables "học sinh học" may result in the following word segmentations (học) (sinh) (học), (học sinh) (học), or (học) (sinh học) (Le et al., 2010). Figure 5.11 shows that each overlap ambiguity

string results in an ambiguity group, therefore, if a graph has k ambiguity groups, there are 2^k segmentations of the underlying phrase (Le et al., 2008).

Due to those ambiguities of word segmentation in Vietnamese, a hybrid approach was adopted from the work of Le et al. (2008). It combined finite-state automata technique, regular expression parsing and the maximal-matching strategy that was augmented by statistical methods to resolve ambiguities of segmentation. The Vietnamese lexicon in use was compactly represented by a minimal finite-state automaton. An input text was first parsed into lexical phrases and other patterns using pre-defined regular expressions. The automaton was then deployed to build linear graphs corresponding to the phrases to be segmented. The application of a maximal-matching strategy on a graph resulted in all candidate segmentations of a phrase. Ambiguity was resolved by a smoothed bigram language model, to choose the most probable segmentation of the phrase.

Our word segmentation module extended vnTokenizer (Le et al., 2008), a tokenizer for Vietnamese texts (Precision = 95%, Recall = 96%). Some modifications for input/output processing as intermediate results of the whole system of VTED were implemented. Some refinements were performed for wrong segmentation in loanwords (e.g. xi-măng - *"ciment" in French* – were segmented as two single words) or proper names (e.g. the first syllable of sentences was combined with succeeding syllables with initial capital letters as a proper name).

5.6.2 Text normalization (vted-normalizer)

As presented in Section 3.2, Chapter 3, the input of real text for a TTS system is messy, e.g. numbers, dates, abbreviations, currency. They 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.

Vietnamese NSWs include numbers; digit sequences (such as telephone numbers, date, time, codes...); abbreviations (e.g. "ThS" for "Thạc sĩ"); words, acronyms and letter sequences in all capitals (e.g. "GDP"); foreign proper names and place names (such as "New York"); roman numerals; URL's and email addresses.

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 by 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" *(two thousands zero hundred and ten)* as a number or a year or a number in an address, or "hai không một không" *(two zero one zero)* as a code or a telephone number.

More systematic cases include the format of "number-number", which can be considered as a date, a range of number or even a score, such as "3-5" can be considered as "mùng ba tháng năm" *(the third of May)* for a date, "từ ba đến năm" *from three to five* for a range of number and "tỉ số là ba năm" *(the score is three five)* for a score; or the format of "number/number", which can be considered as a date or a fraction such as "8/99" "tháng tám năm chín mươi chín" *(May, ninety-nine)* for date and "tám phần chín mươi chín" *(eight over ninety-nine)* for a fraction. In addition, abbreviations also can be ambiguous such as "ĐT" can be expanded to "Điện thoại" *(telephone)*, or "Đội tuyển" *(a selected team in a competition)*.

Normalization model. The normalization model for Vietnamese text from this previous work was refined to an improved one, as shown in Figure 5.12.

First, in this model, an input text was tokenized by blanks and punctuations. The resulting tokens were then filtered through the PRO-SYLDIC including pronounceable Vietnamese syllables (cf. Section 2.7, Chapter 2). Tokens that did not appear in the dictionary were considered as candidate NSWs.

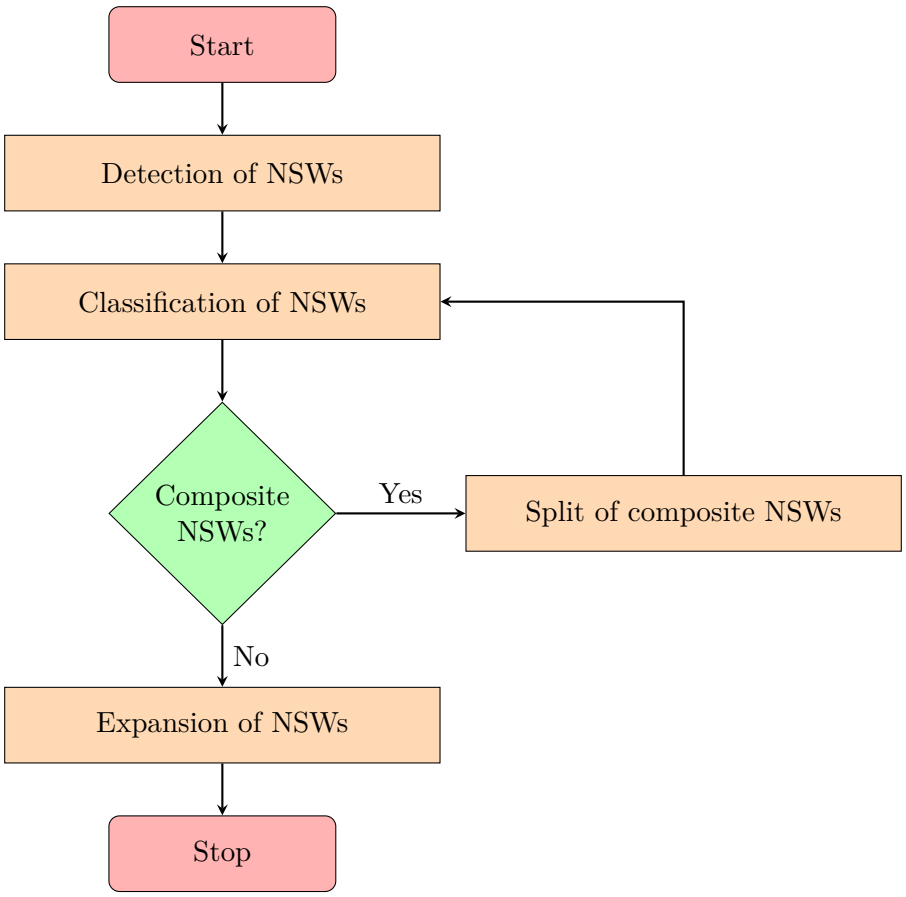


Figure 5.12 – Normalization model for Vietnamese (NSWs: Non-Standard Words).

Second, detected NSWs then are classified into one of categories in Table 5.5 by different strategies for each group. Third, all composite NSWs were split and then re-classified again. Finally, all categorized NSWs were expanded to full words using certain algorithms.

NSW categorization. Based on our previous work (Nguyen et al., 2010), an elaborate categorization of Vietnamese NSW was proposed for VTED, illustrated in Table 5.5. There were three main groups including NUMBERS, LETTERS and OTHERS. The first group, NUMBERS – defined for tokens involving numbers, included 11 categories, such as dates, times, normal numbers, addresses, etc. The second group "LETTERS", classified for NSWs that included only letters, included: loan words, abbreviations, Greek letters, and letter sequence. The last one "OTHER" was defined for remaining NSWs, such as money, measure unit, character sequence, etc.

We have different strategies for each group of above categories. For the NUMBERS group, each category had a typical format; hence was mainly recognized using regular expressions. For ambiguous cases presented above, the decision tree technique was adopted of the classification with the following features: numbers of characters, numbers of digits, two preceding tokens, and two subsequent tokens of NSWs.

Table 5.5 – Vietnamese Non-Standard Word categorization for VTED

Group	Category	Description	Example
NUMBERS	NTIM	time	1:30
	NDAT	date	17/3/87, 1/3/2010, 17/3, 3/2010
	NNUM	number	2009, 70.000
	NTEL	telephone number	0915.334.577
	NDIG	number as digits	Mã số 534
	NSCR	score	Tỉ số là 3-5
	NRNG	range	Từ 3-5 ngày
	NFRC	fraction	34/6, 45,6/145
	NADD	address	số 14/3/2 phố Huế
LETTERS	LOAN	loan word	London, NATO
	LSEQ	letter sequence	ODA, GDP
	LABB	abbreviation	CLB (Câu Lạc Bộ)
	LGRK	Greek letter	I, II, III, IV, VI
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
	NONE	ignored	asscii art...
	COMP	composite	2x2x3, 2*3, VN291
	UNIT	measure unit	m2, m3, kg, %

For the LETTERS group, which consists of strings of alphabetic characters, there was no specific format for each category and the context also might not play an important role for the classification. NSWs including letters were classified as loan words or abbreviations if they were found in the loan word or abbreviation dictionary respectively. Greek letters were identified by two preceding tokens, which were observed in the raw text. Remaining letter NSWs were considered as letter sequences. For the OTHERS groups, regular expressions were adopted for the classification.

NSW Expansion. The final and most important step was to expand NSWs to the full format. Some expanding algorithms were developed for NSW categories in each group. For instance, a cardinal number in Vietnamese could be expanded by following rules: (i) numbers from 0 to 10: 0-hai, 1-một, 2-hai, 3-ba, 4-bốn, 5-năm, 6-sáu, 7-bẩy, 8-tám, 9-chín, 10-mười; (ii) ten numbers (for numbers above ten and below a hundred): tens-"mười" and (e.g. 11-"mười một", 14-"mười bốn", 19-"mười chín"), twenty-"hai mươi", thirty-"ba mươi",.., ninety-"chín mươi"; (iii) no separation (e.g. "and" in English) in the whole part of a number, such as between hundreds and tens, (iv) the expansion of ten numbers: the number of tens (even with zero-"không") followed by the word "mười" *(thousand*; similar for hundreds "trăm", thousands "nghìn", millions "triệu", billions "tỉ", million billions "triệu tỉ", billion billions "tỉ tỉ", ect., (v) few numbers with exceptional/special names: "linh" for zero tens such as e.g. 1001-"một nghìn không trăm linh một" *(*one thousand zero hundred "zero" one), "tư" for four in numbers from 21 to 99 such as 24-"hai mươi tư", 44-"bốn mươi tư", 84-"tám mươi tư". Ordinal numbers in Vietnamese were pronounced in a simple way. They had similar rules to cardinal numbers with the following exceptional/additional rules: (i) starting with the syllable "thứ" for ordinal, (ii) 1-"nhất", 4-"tư".

The full words for abbreviations and loan words could be looked up by the corresponding dictionaries. As presented in Section 3.3, an abbreviation dictionary was manually built with 435 entries including pairs of abbreviations and explanations, such as "ĐHBKHN" was expanded to "Đại học Bách Khoa Hà Nội" *(Hanoi University of Science and Technology)*, "CLB" was expanded to "câu lạc bộ" *club*. Another dictionary was also constructed and composed of 2,821 loan words with their corresponding pronunciation. Several ambiguous full words could be found in the dictionary for abbreviations, such as "TS" can be expanded to "Tiến sĩ" or "Thí sinh". Therefore, a trigram language model with Bayes for probability estimation was used to disambiguate these cases. Other categories could be expanded by quite-well-defined algorithms with certain exceptions. Details of those can be found in Nguyen et al. (2010).

Development of *vted-normalizer* module. Although Nguyen et al. (2010) reported an experiment with a good accuracy (i.e. 98% in NSW classification and 93% in NSW expanding), there was no available software for a complete Vietnamese text normalizer that could integrate into VTED. A *vted-normalizer* module was hence designed based on the improved categorization and model of NSW presented above, and then implemented and integrated into VTED. Due to limited time, the disambiguation of NSW expansion using language models was not implemented.

5.6.3 Grapheme-to-phoneme conversion (vted-g2p)

The G2P Conversion module converted the graphemes of normalized syllables from the previous module into phonemes (i.e. tonophones) with its lexical tone. For instance, "tích" *(to accumulate)* could be transformed to [ti5bk̟5b] with the rising tone 5b, "chanh" *(lemon)* [tɕɛ̆1n̟1] with the level tone 1.

As presented in Section 2.7 – Chapter 2, a pronunciation dictionary was constructed with 21,648 pronounceable syllables with tones (orthography) – PRO-SYLDIC. To build this dictionary, 772 rhymes with tones were combined with 19 initial consonants for a complete list of pronounceable syllables. There did exist many nonexistent syllables in some combinations, however they were useful to transcribe loanwords or newly appeared syllables. For example, a loanword "boa" [ɓwa] in 'tiền boa' from French 'pourboire' *(tip)* appeared sometimes in the real input text although it does not exist in the language.

The *vted-g2p* module was developed and integrated into the VTED system. This module included two main tasks. First, the grapheme of a syllable was looked up in the PRO-SYLDIC dictionary. If there was no entry corresponding to this syllable, the G2P rules will be used. There did exist rules for consonants and vowels/diphthongs using X-SAMPA representation since X-SAMPA can cover the entire range of characters in the IPA. We built a well-defined G2P rules with some exceptional cases. The details of G2P rules are presented in Section 2.5, Chapter 2.

5.6.4 Part-of-speech (POS) tagger

Part-Of-Speech (POS) tagging, also called grammatical tagging or word-category disambiguation, is the problem of assigning each word in a sentence the part of speech that it assumes in that sentence. POS tags provide an important feature for a TTS system, especially with HMM-based approach.

In Section 1.4, Chapter 1, we described a number of characteristics of Vietnamese that distinguish it from occidental languages although Vietnamese text is written in a variant of

the Latin alphabet. For instance, the work of Le et al. (2010) reported that Vietnamese is an inflectionless language in which its word forms never change. Since there is no inflection in Vietnamese, all the grammatical information is conveyed through word order and tool words. The inflectionless characteristic makes a special linguistic phenomenon common in Vietnamese : type mutation, where a given word form is used in a capacity that is not its typical one (a verb used as a noun, a noun as an adjective...) without any morphological change. This leads to the fact that Vietnamese word forms are usually highly ambiguous in their part-of-speech. The same word may be a noun in a context while it may be a verb or a preposition in other contexts. For example, the word yêu may be a noun (the devil) or a verb (to love) depending on context.

Furthermore, Vietnamese is an isolating language, the most extreme case of an analytic language, in which each morpheme is a single, isolated syllable. Lexical units may be formed of one or several syllables, always remaining separate in writing. Although dictionaries contain a majority of compound words, monosyllabic words actually account for a wide majority of word occurrences. This is in contrast to synthetic languages, like most Western ones, where, although compound words exist, most words are composed of one or several morphemes assembled so as to form a single token (Le et al., 2010).

Because of its inflectionless nature, Vietnamese does not have morphological aspects such as gender, number, case... like in occidental languages. Vietnamese words are classified based on their combination ability, their syntactic functions and their general meaning (Le et al., 2010).

As such many difficulties and ambiguities in Vietnamese POS tagging, we adopted *vntagger*, a Vietnamese POS tagger developed in Le et al. (2010) to build the POS Tagging module. Based on the classical principle of maximum entropy, the software yielded a 93.40% overall accuracy and a 80.69% unknown word accuracy on a test set of the Vietnamese treebank. The complete POS tag set of *vntagger* was designed for use in the Viet-TreeBank (Nguyen et al., 2009), as presented in Table 4.3, Section 4.3, Chapter 4.

5.6.5 Prosody modeling

The Prosody Modeling module used information from the previous modules with a prosody model to produce prosodic features. In this module, syntactic information was extracted from syntax trees from the VTParser, or from manually-corrected syntax files.

In our preliminary work, we experimented the ToBI model for extracting ToBI boundary tones for three sentence modes: declarative, exclamative and interrogative sentence. In declarative mode, sentence-internal boundaries were labeled L-H% and sentence final boundaries were labeled L-L%. Interrogative sentences were transcribed with H-H% while exclamative sentences were labeled with L-H% pattern. Phrases terminated by a punctuation mark in the middle of the sentence had a H-L% pattern. Details of these are presented in Section 5.4. Both constituent and dependent rules in our preliminary prosodic phrasing model (Section 4.4, Chapter 4) were also implemented in the Prosody Modeling module for experiment. Actually, these rules were only used for predicting boundaries of prosodic phrases.

The final prosodic phrasing model using syntactic blocks was implemented for two levels: pause appearance and final lengthening (Section 4.6 and 4.7 – Chapter 4). Syntactic blocks were syntactic phrases whose sizes were bounded with a specific number of syllables ("6" for final lengthening and "10" for pause appearance).

Pause appearance was detected using the J48 decision tree of WEKA with the three predictors including POS, syntactic link, and syntactic blocks bounded with 10 syllables. The procedure for identifying these syntactic blocks is presented in Listing 4.1, Chapter 4.

These pauses were then inserted into transcriptions of input text, and modeled as a normal segment in an HMM-based TTS system. Moreover, along with punctuations in the middle of sentences, predicted pauses were also used as the boundaries of prosodic phrases (i.e. intonation ones).

For final lengthening, some new prosodic features for Vietnamese HMM-based speech synthesis were proposed using syntactic blocks bounded with 6 syllables, shown in Table 5.3, Section 5.4. They were number of syntactic blocks in the current sentence, position of the current syntactic block in the current sentence, number of syllables in the current syntactic block, position of the current syllable in the current prosodic block. The procedure for identifying these syntactic blocks is illustrated in Listing B.1, Appendix B.

5.6.6 Feature Processing

This module was adapted and extended from Mary TTS to be suitable for tonal languages including Vietnamese. The Feature Processing module inputs all information from previous modules; process them to build a set of features including contextual features in phoneme, syllable, word, phrase, and utterance level.

The number of the proposed tonophone set was 207, while Mary TTS supported a maximum of 128 phonemes by default (they actually do not design for tonal languages). Therefore, a number of modifications were done in several tens of Java classes due to a complex feature structure with a flexible handling. Furthermore, there were plenty of new Vietnamese contextual features, compared to the existing ones in Mary TTS. For example, at the syllable level, the lexical tones of preceding, current, and subsequent syllables had to be extracted. Prosodic phrase boundaries were identified by not only punctuations in the middle of sentences, but also by the pause prediction model using the three predictors: POS, syntactic link, and syntactic blocks bounded with 10 syllables. A number of new features for final lengthening, presented in Section 5.4, were also extracted for the HMM-based training.

5.7 VTED training voices

Five versions of VTED with different training features or corpus were implemented showing the progress of our work. Several training feature sets were used for the experiment: (i) Basic feature set (ii) ToBI features (iii) new final lengthening features using syntactic blocks having a maximum of 6 syllables.

Table 5.6 summarizes main points of HoaSung and different versions of VTED. All those TTS names in this table were used to represent the results of our perceptual tests in Chapter 6. The three first versions were trained by the existing small corpus, VNSP-Broadcaster, recorded by a broadcaster (92% of 630 sentences): (i) VNSP-VTed1 using the basic feature set plus ToBI features, (ii) VNSP-VTed2 using the basic feature set, and (iii) VNSP-VTed3 using the basic feature set, prosodic phrases were predicted by both constituent and dependent rules and punctuations in the middle of sentences. The two last version of VTED were trained by our new corpus, VDTS, recorded by a non-professional speaker: (i) VDTS-VTed4 using the basic feature set, and (ii) VDTS-VTed5 using the basic feature set plus new final lengthening features, prosodic phrases were predicted by the syntactic-block+link+POS model with the J48 decision tree of WEKA.

In both VNSP-VTed3 and VDTS-VTed5, at the predicted boundaries of prosodic phrases, pauses were inserted into transcriptions of input text, and modeled as a normal segment in an HMM-based TTS system. In the VNSP-VTed3, manually-corrected syntax trees were used

Table 5.6 – HoaSung TTS and different versions of VTED

Synthesis technique	**TTS name**	**Training corpus**	**Training feature set**	**Automatic**
Non-uniform unit selection	VNSP-HoaSung	92% VNSP	Similar to basic features	Yes
HMM-based	VNSP-VTed1		Basic feature set plus ToBI features	Yes
	VNSP-VTed2		Basic feature set	Yes
	VNSP-VTed3		Basic feature set; phrases were identified by syntactic rules	No
	VDTS-VTed4	VDTS	Basic feature set	Yes
	VDTS-VTed5		Basic feature set and final lengthening features using syntactic blocks	Yes

for applying syntactic rules to predict prosodic phrases. All four remaining versions were automatically trained and synthesized. In the VDTS-VTed5, the TTS system use VTParser for an automatic syntactic parsing.

For the sake of comparison, some perception tests were also carried out on our previous TTS system adopting non-uniformed unit-selection synthesis – HoaSung (Do et al., 2011). The training corpus of HoaSung is the same as the Initial VTED-VNSP (92% of VNSP corpus). Feature system of HoaSung had similar basic training features as VTED, cf. Tran (2007b).

These voices and the online system are now available in the corresponding menu of Nguyen (2015b).

5.8 Conclusion

We presented in this chapter a complete architecture of an HMM-based Vietnamese TTS system, VTED. The architecture of VTED composed of three parts: (i) Natural language processing (NLP), (ii) Training, and (iii) Synthesis. From the input text, the NLP part extracts contextual features to provide for both Training and Synthesis phase. In the Training phase, these features are then aligned with speech unit labels and trained with speech parameters (i.e. spectral and excitation) to build context dependent HMMs. In the Synthesis phase, according to a label sequence with these features, a sequence of speech parameters was produced. Finally, a synthetic speech was obtained using these speech parameters and a vocoder.

Contextual features for Vietnamese were chosen at tonophone, syllable, word, prosodic phrase and utterance levels inheriting from other works with an adaption for Vietnamese. Two preceding and two succeeding tonophones were chosen for the phonemic context of the current tonophone due to the stable Vietnamese syllable structure with four elements. There were locative features of the current tonophone, syllable, word or phrase in the current syllable, word, phrase or utterance. The numbers of the two lower levels in the 3-gram model were also considered, e.g. at Word level: number of tonophones in the previous/current/next word, number of syllables in the previous/current/next word. Some prosodic features, such

as POS of previous/current/next word, break index of phonemes, were used. Tone of the previous/current/next syllable is an important feature for Vietnamese. The ToBI boundary tone feature was used for further experiment.

After carefully studying various platforms, Mary TTS was chosen for building VTED due to its ease of use, expandability, and a mixed excitation vocoder. It is an open-source, multilingual Text-to-Speech Synthesis platform and now has a big development community in GitHub. Our work focused more on the NLP part, including text pre-processing and prosody modeling. Several existing results of word segmentation, POS tagging and syntax parsing were adopted and adapted to our system with some refinements. Some modules are newly developed: G2P conversion, text normalization, and prosody modeling. Other modules were inherited from Mary TTS and existing tools. The Feature Processing and Feature Label Alignment module were modified and extended for Vietnamese.

Following the prosodic phrasing model using syntactic blocks proposed in Chapter 4, several new prosodic features for final lengthening were proposed. They included number of syntactic blocks in the current sentence, position of the current syntactic block in the current sentence, number of syllables in the current syntactic block, and position of the current syllable in the current prosodic block. Pause appearance was detected using the J48 decision tree of WEKA with the three predictors including POS, syntactic link, and syntactic blocks bounded with 10 syllables. These pauses were then inserted into transcriptions of input text, and modeled as a normal segment in an HMM-based TTS system. Moreover, along with punctuations in the middle of sentences, predicted pauses were also used as the boundaries of prosodic phrases (i.e. intonation ones).

Five training voices were built using different feature sets and/or training corpus for perceptual evaluations. The three first versions were trained by the existing small corpus, VNSP-Broadcaster: (i) VNSP-VTed1 using the basic feature set plus ToBI features, (ii) VNSP-VTed2 using the basic feature set, and (iii) VNSP-VTed3 using the basic feature set, prosodic phrases were predicted by syntactic rules and punctuations in the middle of sentences. The two last version of VTED were trained by our new corpus, VDTS, recorded by a non-professional speaker: (i) VDTS-VTed4 using the basic feature set, and (ii) VDTS-VTed5 using the basic feature set plus new final lengthening features, prosodic phrases were predicted by the syntactic-block+link+POS model with the J48 decision tree of WEKA. In both VNSP-VTed3 and VDTS-VTed5, at the predicted boundaries of prosodic phrases, pauses were inserted into transcriptions of input text, and modeled as a normal segment in an HMM-based TTS system. In the VNSP-VTed3, manually-corrected syntax trees were used for applying syntactic rules to predict prosodic phrases. All four remaining versions were automatically trained and synthesized. In the VDTS-VTed5, the TTS system use VTParser for an automatic syntactic parsing.

Chapter 6

Perceptual evaluations

Contents

6.1 Introduction

VTED was designed and implemented to be a complete HMM-based Text-To-Speech system for Vietnamese (cf. Chapter 5). The final version of VTED was trained with VDTS – a new designed and recorded corpus (cf. Chapter 3), and a proposed prosodic phrasing model using syntactic blocks (cf. Chapter 4). As a result, five voices were constructed using different training corpora and feature sets, presented in Section 5.7, Chapter 5. These voices and the online system are now available in the corresponding menu of Nguyen (2015b).

In this chapter, the perceptual evaluations of five versions of VTED are described showing the progress of our work. Several types of perception tests were chosen for different purposes: (i) the Mean Opinion Score (MOS) test for general naturalness (ii) the intelligibility test (iii) the tone intelligibility test in context, and (iv) the pairwise preference test for assessing the prosodic models. In the intelligibility test, subjects were asked to write down texts of utterances, which were presented in a Latin square matrix. In the tone intelligibility test, subjects were asked to choose the most likely syllable they had heard among a group of syllables bearing different tones in an utterance. In the pair-wise preference test, subjects listened to stimuli composed of two sounds corresponding to synthetic voices, separated by a "beep" sound.

For the sake of comparison, some perception tests were also carried out on our previous TTS system adopting non-uniformed unit-selection synthesis – HoaSung (Do et al., 2011). The training corpus of HoaSung is the same as the Initial VTED-VNSP (92% of VNSP corpus). Feature system of HoaSung had similar basic training features as VTED, cf. Tran (2007b).

All tests were performed in a soundproof room at MICA, Hanoi, Vietnam. All the participants were from the Northern Vietnam, and had lived for a long time in Hanoi. Subjects were 20-35 years old and reported normal hearing and vision. The test corpora for each test were chosen or designed in parallel with the training corpora. In other words, the test corpora were not included in the training.

The experimental results will be presented using bar plots with mean values and confident intervals, i.e. the interval estimates of a population parameter. Confident intervals were calculated from observations, to indicate the reliability of an estimate. Results of perception tests were statistically analyzed and thoroughly discussed. An alpha level of 0.05 was adopted in the analysis of the test results.

In this work, a test tool, **VEVA** (Vted EVAluation tool), was specially designed for each perception test and implemented using Java. This was a portable tool, which could be deployed in any operating system since all specific-platform aspects were considered during the design and implementation. The output data of this tool was stored in XML files with a well-designed structure, facilitating its extensibility and portability. This test tool was deployed and used in both Windows and Mac OS X for the above tests.

Using VEVA, stimuli were randomly presented to subjects, with several options randomness. For example, the order of the two voices in a stimulus in the pair-wise preference test was random, meanwhile the groups and the options in each group in the tone intelligibility test were also generated randomly to present to participants. In the intelligibility test, the voice order in the Latin square matrix was randomly triggered for each subject. The Graphical User Interface (GUI) of the tests in VEVA was designed with a full-screen mode, which prevented subjects from getting distracted by other objects (e.g. icons, applications, and bars) on the screen. Demonstrations of some GUI screens of VEVA are illustrated in Section C.3 in Appendix C for various perceptual tests.

6.2 Evaluations of ToBI features

The purpose of these tests is to evaluate the role of ToBI features in HMM-based TTS for the Vietnamese language. The two versions of VTED using the existing corpus (VNSP) were built to perform these tests: (i) VNSP-VTed1: VTED trained with the basic training set and ToBI features, and (ii) VNSP-VTed2: VTED trained with the basic training set. A subjective evaluation, i.e. pair-wise preference test, was carried out on these two versions first. An objective test was then observed for further investigation on the results of subjective test.

6.2.1 Subjective evaluation

The subjective evaluation was performed through the pair-wise preference by 16 subjects (8 females) testing the influence of the ToBI features feature on the quality of the synthetic voice. 48 sentences (i.e. 8% of VNSP corpus) were synthesized by two versions of VTED: (i) VNSP-VTed1 (With ToBI) and (ii) VNSP-VTed2 (Without ToBI). To facilitate subjects to compare both systems, long sound files of both voices were split into 92 shorter ones with a length ranging between 5 and 13 syllables. As a result, subjects listened to 92 stimuli, composed of two sounds corresponding to synthetic voices, separated by a "beep" sound. The order of the two voices in each pair and the order of utterances were random.

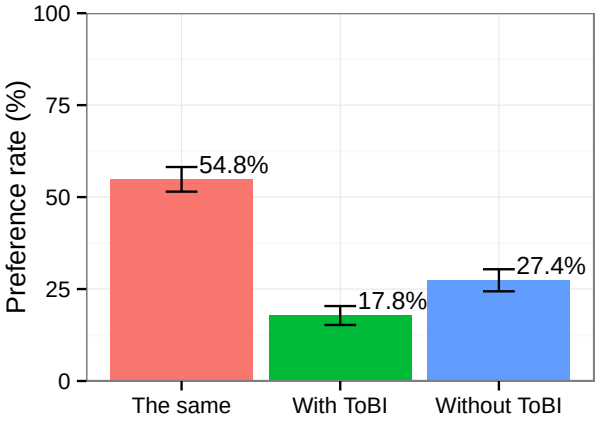


Figure 6.1 – Preference rate of VNSP-VTed1 (With ToBI) and VNSP-VTed2 (Without ToBI).

The experiment results plotted in Figure 6.1 show that subjects perceived the performances of both system as being "The same" in about 55% of the pairs, and the VNSP-VTed2 (Without ToBI) was preferred in 27%, while the VNSP-VTed1 (With ToBI) was preferred in 18% of the pairs – thus about a 10% preference for the VNSP-VTed2 voice. To further analyze the factors that may have affected the perception of the synthetic voice's intonation – i.e. boundary modes (DE, EX, IN or DM as in Chapter 5) and lexical tones of last syllables (1-4, 5a, 5b, 6a or 6b) of sounds in each stimulus, a statistical analysis was run on the results, expressed on a three-point scale of preference.

Figure 6.2 illustrates the preference rate by both boundary modes and lexical tones of last syllables, on the 3-point scale: VNSP-VTed2 is a positive preference mark (+1), while an answer "The-same" is neutral (0), and a preference for the VNSP-VTed1 is a negative mark (-1). This polarity for the scale was chosen after the observed preferences marked by

listeners. Since declarative sentences were the most common in the test corpus, they provided all lexical tones for both phrase positions: middle (DM) or end (DE) of the sentence; while only tones 1, 5a were presented in exclamative (EX) and tones 1, 2, 4, 5b in interrogative (IN) sentences.

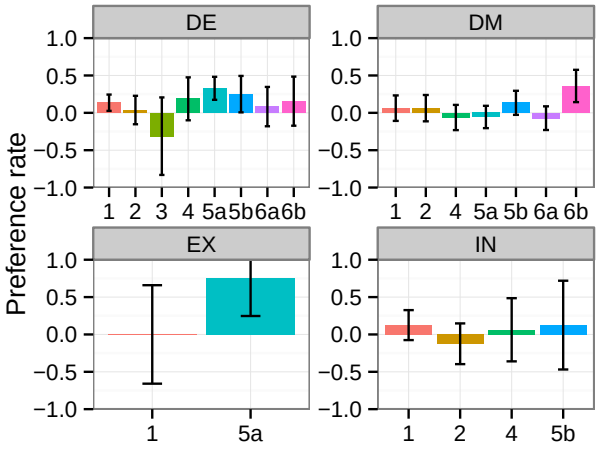


Figure 6.2 – Preference rate by lexical tones and boundary modes with a 3-point scale: (+1) VNSP-VTed2 (Without ToBI), (0) The-same, and (+1): VNSP-VTed1 (With ToBI).

A two-factorial ANOVA was run on the results to see if there was a difference in the use of the 3-point scale, according to the Boundary mode (4 levels) and Lexical tone of the last syllable (8 levels). The result of this analysis, presented in Table 6.1, shows that both factors and their interactions had significant effects ($p<0.05$). The two-way ANOVA indicated there was significant interaction between the effects of the boundary modes and the lexical tones of the last syllables on the subjects' perception.

Table 6.1 – Anova results of pair-wise comparison test.

Factor	df	df error	F	p	η^2
Boundary mode	3	1,451	6.537	0.00022	0.013
Last tone	11	1,451	4.697	0.00003	0.022
Boundary mode:Last lexical tone	22	1,451	3.075	0.00071	0.021

Concerning the impact of tones of the last syllables, for the falling tone (2), curve tone (3) and drop tone in sonorant-final syllables (6a), most subjects did not find any difference between the two voices; while the voice trained without ToBI features was preferred for the level tone (1), rising tones (5a), (5b) and drop tone in obstruent-final syllables (6b). Conversely, the broken tone (4) was preferred in VNSP-VTed1, despite its sparseness in the test corpus.

6.2.2 Objective evaluation

The perception test results show that the synthetic voice trained with ToBI features (VNSP-VTed1) was less preferred. To provide an explanation for that, some observations were made on the preferred VNSP-VTed2 signals.

The two main problems found in the voice with ToBI features were: (i) the discontinuity in spectrum and (ii) the unexpected voice quality. This might have made subjects uncomfortable

(a) Without ToBI

(b) With ToBI

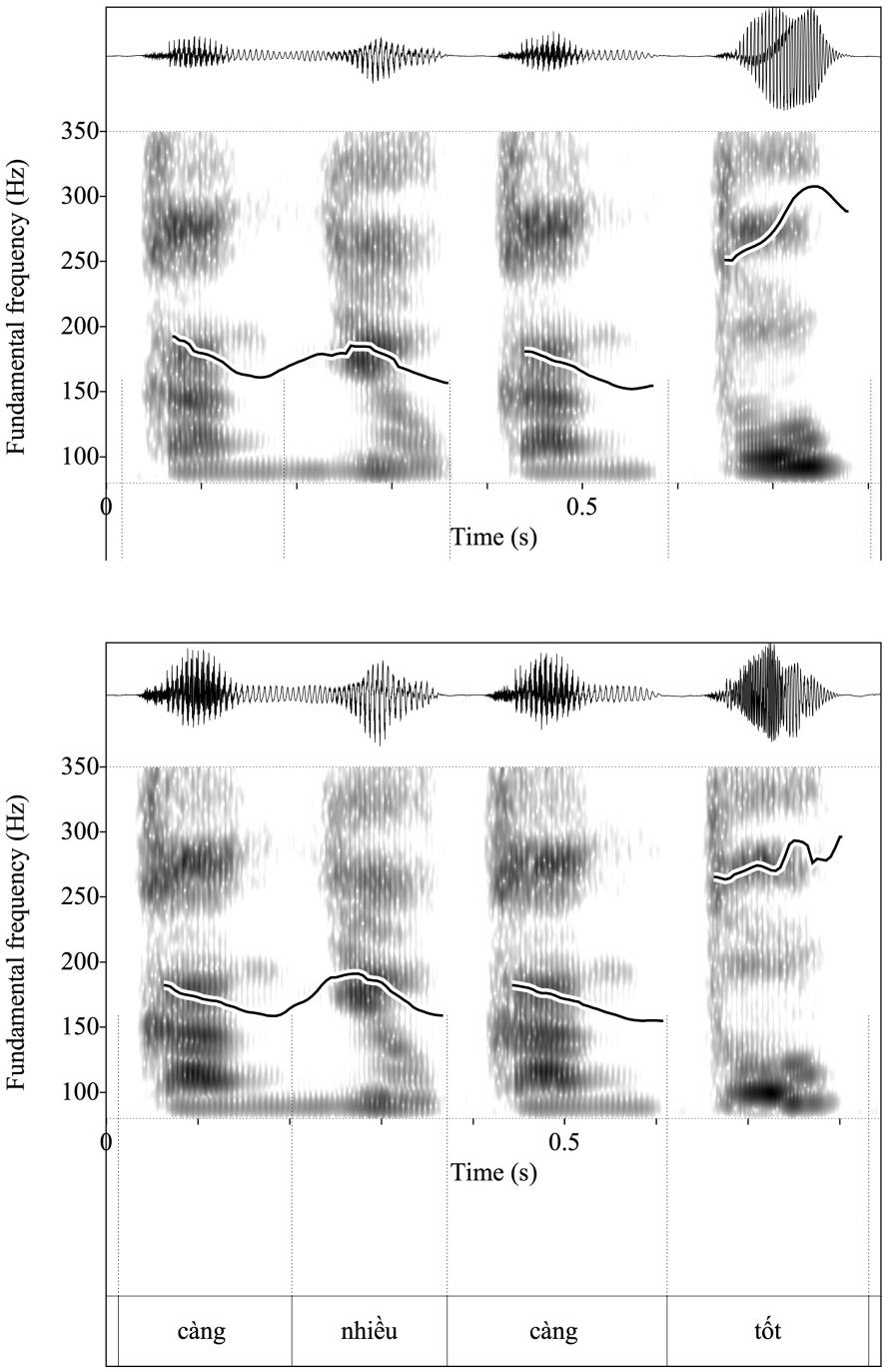
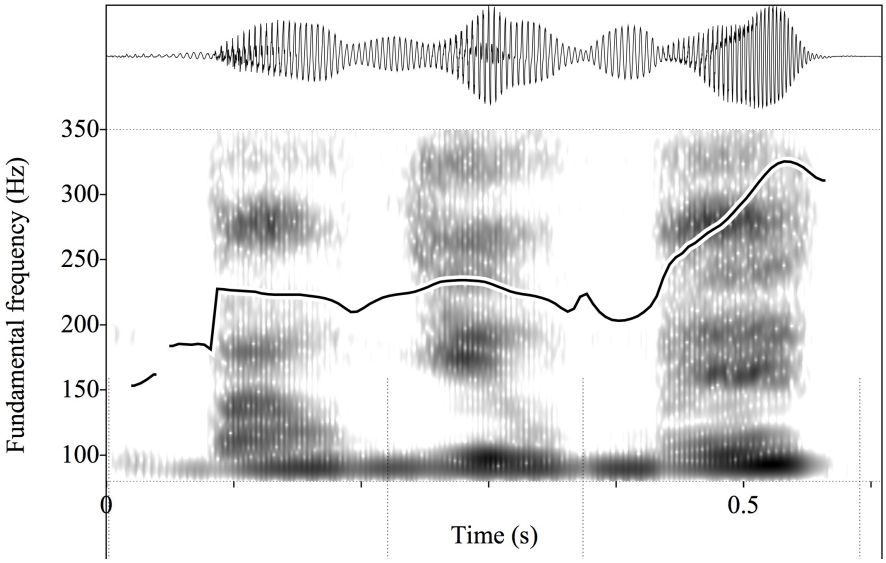


Figure 6.3 – Discontinuity in spectrum and F0 in (b) VNSP-VTed1 (With ToBI) compared to (a) VNSP-VTed2 (Without ToBI) of of "tốt" [tot-5b] *(good)* in "... càng nhiều càng tốt" [kaŋ-1 ɲiew-2 kaŋ-1 tot-5b] *(as much as possible)*.

(a) Without ToBI



(b) With ToBI

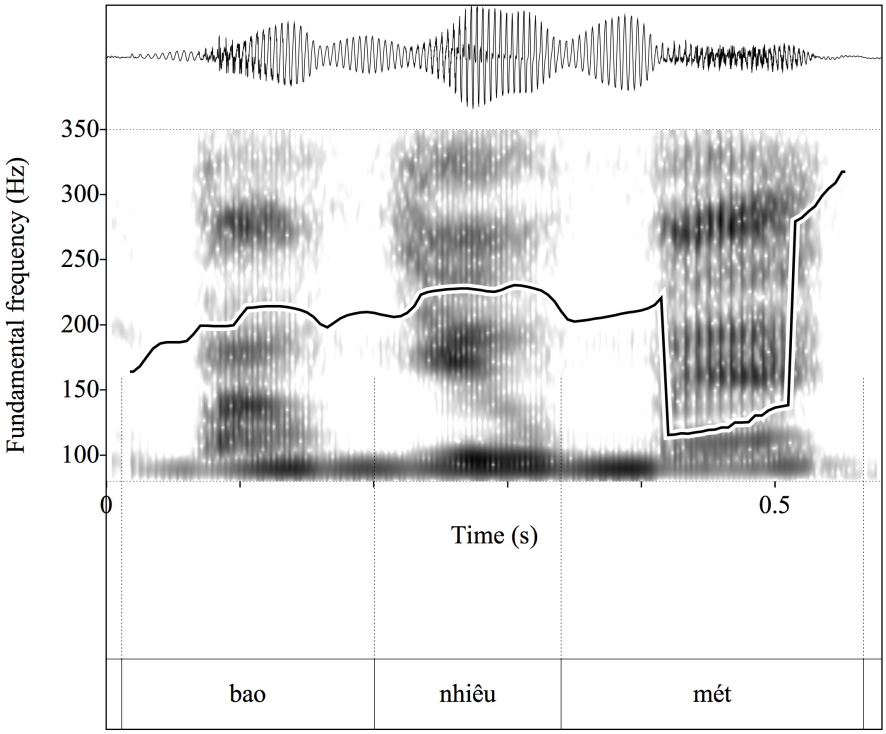


Figure 6.4 – Unexpected voice quality in (b) With ToBI compared to (a) Without ToBI of "mét" *(meter)* [mɛt-5b] in "bao nhiêu mét" [baw-1 ɲiew-1 mɛt-5b] *(how many meters)*.

when hearing the sounds, and have led them to prefer the voice trained without the ToBI features.

Figure 6.3 shows an example of the discontinuity in the spectrum of the (a) "VNSP-VTed1" (With ToBI) of a phrase "càng nhiều càng tốt" *(as much as possible)*, compared to the synthetic speech, and (b) "VNSP-VTed1" (Without ToBI). The intonation pattern "L-L%" was applied to this declarative sentence, following the ToBI rules presented in Chapter 5. However, the F0 contour of the syllable bearing rising tones (5a, 5b) normally raises from the beginning to the end of the syllable. Nevertheless in this case the last syllable "tốt" [tot-5b] *(good)* of the phrase synthesized with ToBI model bearing the rising tone (tone 5) seemed to be flat and discontinuous.

An example of the unexpected voice quality of the VNSP-VTed1 (With ToBI) is illustrated in the Figure 6.4. The sentence "Nhà này rộng bao nhiêu mét?" *(How many square meters is this house?)* used the intonation pattern "H-H%" for an interrogative sentence. The F0 contour of the last syllable "mét" [mɛt-5b] bearing the rising tone (originally high register) was traditionally raised, but in this case we found a phenomena of glottalization in the syllable. The F0 contour of this syllable looked like the F0 contour of the broken tone (4); meanwhile the F0 contour of this syllable for the "VNSP-VTed2" (Without ToBI) maintained the traditional form of the rising tone in obstruent-final syllables (5b).

Although the test corpus did not cover all cases of boundary modes as well as tones of last syllables, we assumed that ToBI features in general did not help in ameliorating the quality of Vietnamese TTS systems. As a result, these ToBI features were removed from the training feature set of the latter versions of VTED (i.e. from version 2 to 5). The experimental results showed the need for more efforts in intonation modeling for Vietnamese TTS, which should take care of the lexical tones and other prosody cues.

6.3 Evaluations of general naturalness

The MOS test was chosen for this evaluation, which allowed us to score and compare the global quality of TTS systems with respect to natural speech references. Subjects were asked to assess the speech they had heard. The question presented to subjects was *"How do you rate the naturalness of the sound you have just heard?"*. Subjects could choose one of following five options (5-scale): (5) Excellent, very natural (human speech) (4) Good, natural (3) Fair, rather natural (2) Poor, rather unnatural (rather robotic) and (1) Bad, very unnatural (robotic).

Stimuli were randomly and separately presented only once to subjects. Each stimulus was an output speech of a TTS system or a natural speech for a sentence. The numbers of syllables of sentences ranged from 2 to 30, hence the speech samples lasted between 1 and 24 seconds. Some examples of sentences in the MOS test can be found in Table C.2, Appendix C.

The perceptual evaluations of the general naturalness were carried out on different versions of VTED with a MOS test. The first MOS test was done to assess the quality of the initial version of VTED trained with the VNSP corpus ("VNSP-VTed1") with a natural speech reference. Both initial and final versions of VTED were chosen in the final MOS test with two natural speech references of training corpora. We assumed that this test could help to see the general progress of our work and proposals.

6.3.1 Initial test

This test was the first of our test series to evaluate the naturalness of the first version of VTED ("VNSP-VTed1") with a natural speech reference. 18 subjects (9 females) participated in the study. This test was also carried out on our previous TTS system adopting non-uniformed

unit-selection synthesis - HoaSung (Do et al., 2011) and the same training corpus as VNSP-VTed1. About %8 of VNSP corpus, i.e. 48 sentences, was randomly extracted for this test. Hence a total of 144 stimuli were prepared in the test corpus. Since the text content of these sentences in VNSP was already marked by commas corresponding to perceived pauses in the natural speech corpus, synthetic stimuli of the two systems (i.e. VTed1 and HoaSung) had a quite-good rhymth.

The experiment results plotted in Figure 6.5 show that the quality of VTED was rather good, 0.81 point higher than HoaSung on a 5-point MOS rating scale, but still clear distinguishable 1.21 point lower than the natural speech.

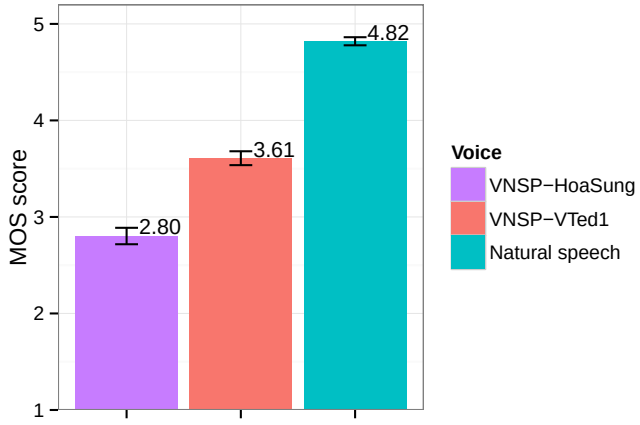


Figure 6.5 – Score of naturalness (MOS Test) of initial HMM-based TTS system VTED, non-uniformed unit-selection TTS system HoaSung, with a natural speech reference.

A two-factorial ANOVA was run on the results. The two factors were the TTS system (4 levels) and the Sentence (48 levels). All factors and their interactions had significant effect ($p<0.05$). The TTS system factor alone explained an important part of the variance, about 63% (partial $\eta^2 = 0.63$), while the Sentence factor and their interaction explained only about 15% each. A post-hoc Tukey test showed that each TTS system received significantly different mean scores ($p<0.05$).

Table 6.2 – Anova results of MOS test for initial VTED versions

Factor	df	df error	F	p	η^2
SystemID	2	2,162	1877.07	0.0000	0.63
Sentence	47	2,162	8.77	0.0000	0.16
SystemID:Sentence	94	2,162	4.19	0.0000	0.15

6.3.2 Final test

This final MOS test aimed at evaluating the general quality progress of the final over the initial version of VTED. 21 subjects (11 females) participated in this test. Since there were two training corpora, i.e. the old corpus VNSP, 16kHz by a professional broadcaster and the new corpus VDTS, 48kHz by a nonprofessional speaker; four voices were used in this test: (i) two synthetic ones: initial version VNSP-VTed1 (old corpus), and final version VDTS-

VTed5 (new corpus and new prosodic phrasing model using syntactic blocks), and (ii) two natural ones recorded by speakers of the VNSP and VDTS corpus, respectively. There were 40 sentences in the test corpus hence totaling 200 stimuli of those 4 voices. The text content of these sentences were kept as the original representation, without an intentional insertion of commas.

Figure 6.6 illustrates the results of this experiment. The quality of the final version of VTED with the new corpus (VDTS-VTed5) progressed by about 1.04 point higher than the initial version with the old corpus (VNSP-VTed1). The gap between the final version of the synthetic voice and the natural voices was also much improved, by about 0.5 point on a 5-point MOS scale.

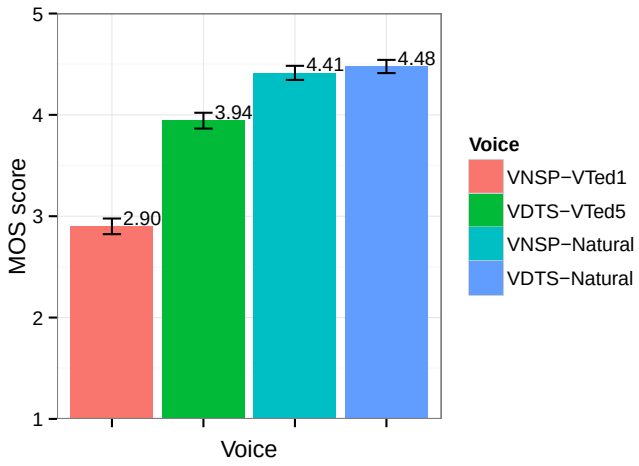


Figure 6.6 – Score of naturalness (MOS Test) of initial and final versions of VTED, and two natural voices.

A two-factorial ANOVA was run on the results, illustrated in Table 6.3. The two factors were the TTS system (3 levels) and the Sentence (40 levels). All factors and their interactions had significant effect ($p<0.05$). The TTS system factor alone explained an important part of the variance, about 40% of the variance (partial $\eta^2 = 0.40$), while the Sentence factor and the interaction explained only about 2% and 7% each.

Table 6.3 – Anova results of MOS test for initial and final VTED versions.

Factor	df	df error	F	p	η^2
System	3	3,200	702.72	0.0000	0.40
Sentence	39	3,200	1.96	0.0004	0.02
System:Sentence	117	3,200	2.04	0.0000	0.07

Tukey's Honest Significant Difference (Tukey multiple comparisons of means) with 99% family-wise confidence level was run on the results to discover the differences among the means of the "System" level. All TTS systems received significantly different mean scores (the p-value after adjustment for the multiple comparisons < 0.05) except for the difference between the two natural voices, i.e. "VNSP Natural" and "VDTS Natural" (p adj = 0.387). The two natural voices are perceived almost equivalently.

6.3.3 Discussion on the two tests

As presented, the test corpus of the initial evaluation was extracted from VNSP, where the text content was already marked by commas corresponding to perceived pauses in the natural speech corpus. Whereas, the text content of sentences in the final test were kept as the original representation, without an intentional insertion of commas. This might affect the absolute scores in these tests.

The MOS score of the first VTED was 3.61 in the initial test, while it was 2.90 in the final test. We found three main reasons for these scores. First, the intentional insertion of commas in the initial test made the first VTED to have a better rhythm. Second, there was a "worse" synthetic speech, i.e. HoaSung, in the initial test; and the subjects tended to relatively classify the voices they heard. Finally, the two tests were conducted in different time (about 2.5 years apart), and by different subjects.

However, the relative gaps between voices in both tests remained comparable. The first version of VTED scored about 1.2 point in the initial test, and about 1.5 point in the final test, lower than the natural speech. The natural voice in the final test was about 0.3 point lower than the one in the initial test. It turned out that participants in the final test gave "stricter" and more "sensitive" rates than in the first one. Although the subjects gave scores in MOS tests, we believed that this kind of perceptual test only provided a relative assessment among the voices in a specific test. The gaps or comparisons between voices in MOS test were reliable and stable.

6.4 Evaluations of general intelligibility

This type of perception test allowed us to evaluate and compare the intelligibility of the TTS system with respect to a natural speech reference. Subjects were asked to write down the utterances they had heard. They could listen to samples once, twice, or three times. The reference and written texts were transcribed for comparison. Error rates or similarity of the written text over the reference text could be calculated by different methods to assess the intelligibility of various voices.

6.4.1 Measurement

To measure the error rate of this test, we adopted the approximate (or fuzzy) string matching method (Navarro, 2001). The Damerau-Levenshtein algorithm and some improvements in dynamic programming were implemented in this work to calculate the distance between any two strings. This distance was measured in terms of the total number of primitive operations (i.e. edit operations) necessary to convert the string into an exact match. Some examples of edit operations including insertion, deletion, substitution and transposition are presented bellow.

- Insertion: a c → a **b** c
- Deletion: a **b** c → a c
- Substitution: a **b** c → a **d** c
- Transposition: a **b** **c** → a **c** **b**.

Error rates were calculated based on the metric proposed in the work of Soukoreff and MacKenzie (2001) in order to normalize the distance between two strings A and B, shown in

Equation 6.2, by the maximum length of these strings. This implementation was applied at syllable, tone and phoneme levels.

$$Distance(A, B) = \sum_{i} N(i) \tag{6.1}$$

$$ErrorRate(A, B) = \frac{Distance(A, B)}{Max(\mid A \mid, \mid B \mid)} \tag{6.2}$$

where

- i: covers insertion, deletion, substitution and transposition
- $N(i)$: Number of edit operation i
- $Distance(A, B)$: The distance between string A and B
- $Max(\mid A \mid, \mid B \mid)$: maximum length of two strings A and B.

6.4.2 Preliminary test

36 sentences with a length ranging from 8 to 20 syllables were extracted from e-newspapers for the test corpus. With two voices including the first version of VTED (VNSP-VTed1) and a natural speech reference, 72 stimuli were presented to 18 subjects (9 females). We first carried out this test in a simple method, in which all stimuli were randomly presented to subjects without any special design. This implied that participants had to listen to and write down two utterances of the two voices with the same content.

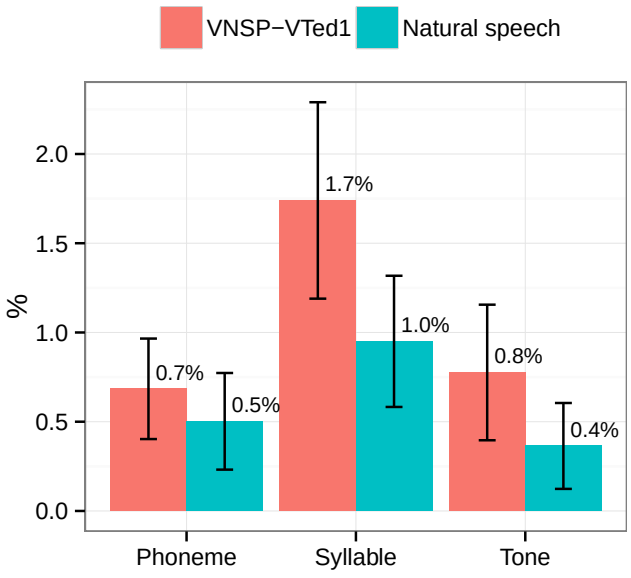


Figure 6.7 – Error rates of intelligibility in utterance elements.

The error rates of VTED and natural speech are illustrated in Figure 6.7. VTED diverged from natural speech from 0.2% - 0.9%. The error rates of VTED in general did not exceed 2%. This preliminary result shows that the first version of VTED was intelligible.

However, as mentioned above, in this preliminary test, subjects were asked to write duplicate text content corresponding to the two test voices. We supposed that participants might have remembered a part or the whole content of the utterances in the second listening, especially for the case in which the natural speech was heard first. This problem might have affected to final scores of TTS systems.

6.4.3 Final test with Latin square

In order to address the issue of duplicate contents of stimuli, we adopted the Latin square design (Cochran and Cox, 1992) for the final test. In this test, there were three voices including a natural speech reference and two versions of VTED: the first version (VNSP-VTed1) and the last one (VDTS-VTed5). With the Latin square design, each subject listened to one third of the utterances per voice, without any duplicate content.

	Sentence 1	**Sentence 2**	**Sentence 3**	**...**
Subject 1	1	2	3	...
Subject 2	2	3	1	...
Subject 3	3	1	2	...
...	...	...	...	...

Table 6.4 – The Latin square design for three voices (1, 2, 3).

The process to choose which utterance of which voice to present to a specific subject followed a Latin square matrix 3x3 as seen in Table 6.4, where "1" is the natural speech, "2" is VNSP-VTed1 and "3" is VDTS-VTed5. For instances, the first subject listens to the sentence "1" of the voice "1", the sentence "2" of the voice "2" and the sentence "3" of the voice "3"; while the second subject listens to the sentence "1" of the voice "2", the sentence "2" of the voice "3" and the sentence "3" of the voice "1", etc. Hence, each sentence was listened to by one third of the subjects and each subject listened to one third of the utterances per voice. The number of subjects and sentences must be a multiple of "3" (i.e. number of voices).

The final test followed the Latin square (3x3) design for presenting stimuli to 24 subjects (14 females). In this test, 108 sentences with a length ranging from 8 to 20 syllables were chosen from e-newspapers, making 324 utterance stimuli (3 systems x 108 sentences). Each subject only had to listen to and write text contents for 108 utterances including 36 sentences of the natural speech, 36 of VNSP-VTed1, and 36 of VDTS-VTed5. The order of voices in designing the Latin square matrix and all chosen stimuli were randomly presented to subjects. Some examples of this test corpus can be found in Table C.3, Appendix C.

The error rates of the three voices, calculated as Equation 6.2, are illustrated in Figure 6.8. At the syllable-level, the error rate of the first version of VTED, VNSP-VTed1, was 14.3%, that is about 12.0% higher than the natural speech. This first version diverged by about 5.8-7.5% from the natural speech at lower levels, i.e. tone and phoneme. The perceptual results showed that the intelligibility of the final version of VTED, VDTS-VTed5, approached that of the natural speech. The gap between VDTS-VTed5 and the natural speech was only between 0.4% and 1.4%. It turns out that the last version of VTED advanced considerably in terms of intelligibility to the first version.

We investigated the results of the final test on edit operations, i.e. representing error manipulations. There was no transposition error but errors existed in three other operations. Figure 6.9 shows the rates of the edit operation errors for the three voices at three levels:

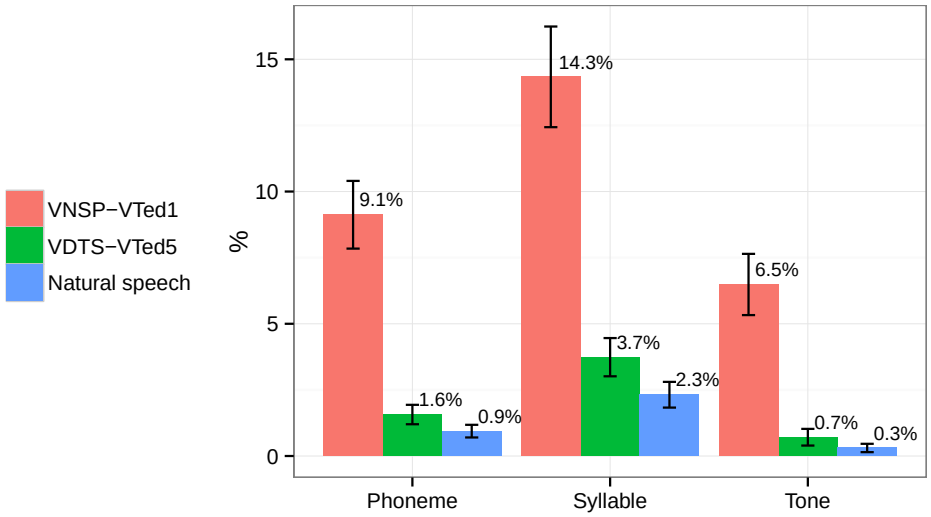


Figure 6.8 – Error rates of initial, final VTED and a natural speech at phoneme, tone and syllable levels. The test was designed based on Latin square matrix 3x3.

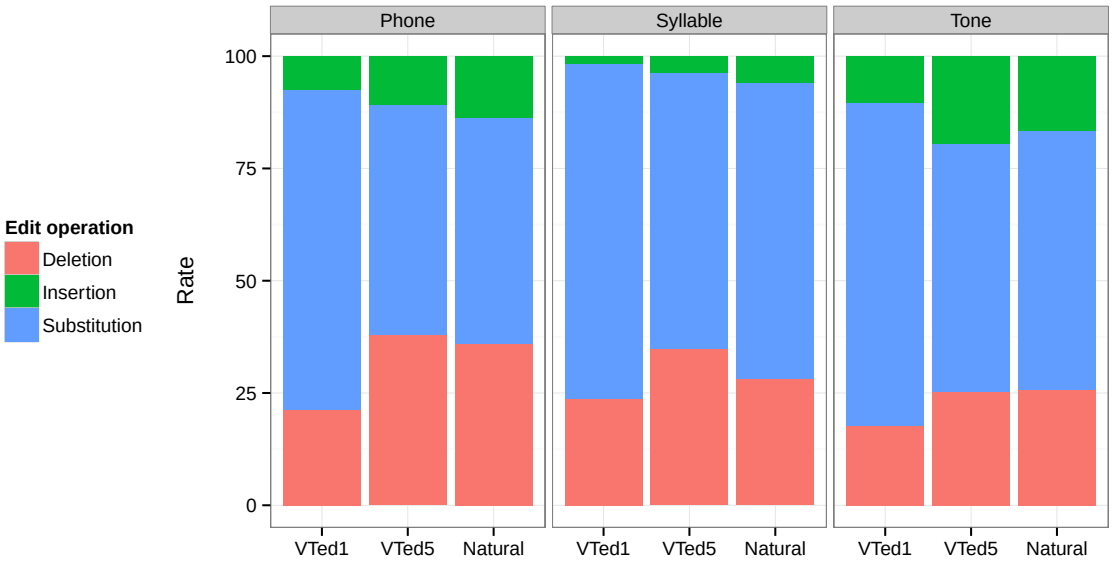


Figure 6.9 – Edit operations of initial, final VTED and a natural speech at phoneme, tone and syllable levels. The test was designed based on Latin square matrix 3x3.

phoneme, tone and syllable. Substitution errors occupies about 55-75% of all the errors, i.e. the largest portion among the three edit operations; while insertion errors made up for only a minor part. The proportions of deletion, substitution and insertion operations of the last version of VTED, VDTS-VTed5, were similar to those of natural speech. In contrast, the initial version of VTED tended to have more substitutions than the natural speech and the last one.

6.5 Evaluations of tone intelligibility

This type of perception tests allowed us to evaluate and compare the tone intelligibility in context of TTS systems with respect to a natural speech reference. This test was also carried out on the initial and final VTED versions.

6.5.1 Stimuli and paradigm

In this test, groups of meaningful sentences with the same syllables and syllable order, diverging by only one tone, were prepared. Proper nouns were used in cases where tone variations of certain syllables would not be meaningful otherwise. Subjects were asked to choose the most likely syllable they heard among a group of syllables bearing different tones in an utterance. Three examples of these groups are presented in the Example 8.

Example 8 Examples of sentences with same ordered syllables, different from only one tone

1. Ở đây có buôn bán ... không? (Do you sell ... here?)
   - dê ([ze-1] – level tone 1, means "goats")
   - dễ ([ze-4] – broken tone 3, means "easily")
   - dế ([ze-5a] – rising tone 5a, means "crickets")
2. Mỗi tối, bác sĩ ... thường đến hỏi thăm các bệnh nhân (Every evening, doctor ... usually visits her patients)
   - Thuỷ ([t^hwj-3] - a person name with curve tone 4)
   - Thuỳ ([t^hwj-2] - a person name with falling tone 2)
   - Thuý ([t^hwj-5a] - a person name with rising tone 5a)
   - Thuỵ ([t^hwj-6a] - a person name with drop tone 6a)
3. Thế này mà nhà cô còn ... gì nữa? (In this situation, why does your family still ...?)
   - tiếc ([tiek-5b] - rising tone 5b, means "regret")
   - tiệc ([tiek-6b] - drop tone 6b, means "have a party").

The first example shows a group of sentences in Vietnamese: "Ở đây có buôn bán dê/dễ/dế không?" *(Do you sell goats/easily/crickets here?)*. In these sentences, the 6^{th} syllable can be one of dê [ze-1] with level tone (1), dễ [ze-4] with broken tone 3, or dế [ze-5a] with rising tone 5a. All these sentences are mostly equal in terms of their frequencies and meanings in Vietnamese.

The second one is an example of using person names: "Mỗi tối, bác sĩ Thuỷ/Thuỳ/Thuý/Thuỵ thường đến hỏi thăm các bệnh nhân" *(Every evening, doctor [t^hwj-3]/[t^hwj-2]/[t^hwj-5a]/[t^hwj-6a] usually visits her patients)*. For the syllable [t^hwj], four possible tones can be applied to a person name: broken tone [t^hwj-3], falling tone [t^hwj-2], rising tone [t^hwj-5a] and drop tone [t^hwj-6a]. These person names are common, though the last one is less frequent but not rare.

The last one illustrates examples of obstruent-final syllables whose final consonants ended in an unreleased oral stop /p t k/. As presented in Chapter 2, only rising (5b) and drop (6b)

Table 6.5 – Coverage of tone pairs of test corpus for the tone intelligibility test

Tone types	**(1)**	**(2)**	**(3)**	**(4)**	**(5a)**	**(5b)**	**(6a)**	**(6b)**
Level (1)	-	8	10	8	10	-	11	-
Falling (2)	8	-	9	12	10	-	9	-
broken (4)	10	9	-	8	9	-	10	-
curve (3)	8	12	8	-	10	-	10	-
Rising (5a)	10	10	9	10	-	-	11	-
Rising (5b)	-	-	-	-	-	-	-	11
Drop (6a)	11	9	10	10	11	-	-	-
Drop (6b)	-	-	-	-	-	11	-	-

tones can appear in obstruent-final syllables. As a result, there were always two options in groups of obstruent-final syllables bearing those tones, e.g. "tiếc" [tiek-5b] *(regret)* and "tiệc" [tiek-6b] *(have a party)*.

The test corpus was designed by taking into account the balance of tone pairs. A total of 130 sentences in 40 groups were ultimately designed. The coverage of tone pairs is illustrated in Table 6.5. The chosen test corpus had a good balance of 8-12 examples for each tone pair. As mentioned, it is impossible to combine obstruent-final and sonorant-final syllables in one sentence group. As a result, there were two types of sentence groups: (i) target syllables that were sonorant-final syllables (for the tones 1-4, 5a, 6a), e.g. the two first group in Example 8, and (ii) target syllables that were obstruent-final syllables (for the tones 5b, 6b). Rising tones and drop tones in obstruent-final syllables (5b and 6b) were always in the same group for comparison, e.g. the last group in Example 8. More detail of these examples and others can be found in Table C.1, Appendix C.

Each stimulus was an output speech of a TTS system or a natural speech for a sentence. Groups, options in each group, and stimuli were randomly presented only once or twice to subjects.

6.5.2 Initial test

In this evaluation, subjects had to choose the most likely syllable they had heard among syllables differing only in tones, or "None of the above options" if they could not find a suitable answer. Subjects had the option to listen to each simile once more after their first attempt. However, they had to indicate their decision after each listening so that we could record their responses between the two trials.

Although the test corpus was designed with 130 sentences in 40 groups, this test was performed with 129 sentences due to a carelessness. With the two voices of the initial version of VTED (VNSP-VTed1) and a natural speech, the number of stimuli presented to subjects was 258 utterances.

The correct rates of this experiment are illustrated in Figure 6.10. The results were from 15-18% higher for utterances needing a single listening (once) than for those needing two listening repetitions (twice). The synthetic speech showed a decrease of about 23% correct rate to the natural speech in the global result. The proportion of "None of the above options" selection was minor, from 0.0 to 0.7% for natural speech, and from 1.0 to 3.4% for "VNSP-VTed1".

The correct rates by tone types in Figure 6.11 show that all tones in the natural speech were nearly 100% perceived in the same context, except for the falling tone (2). It appears that

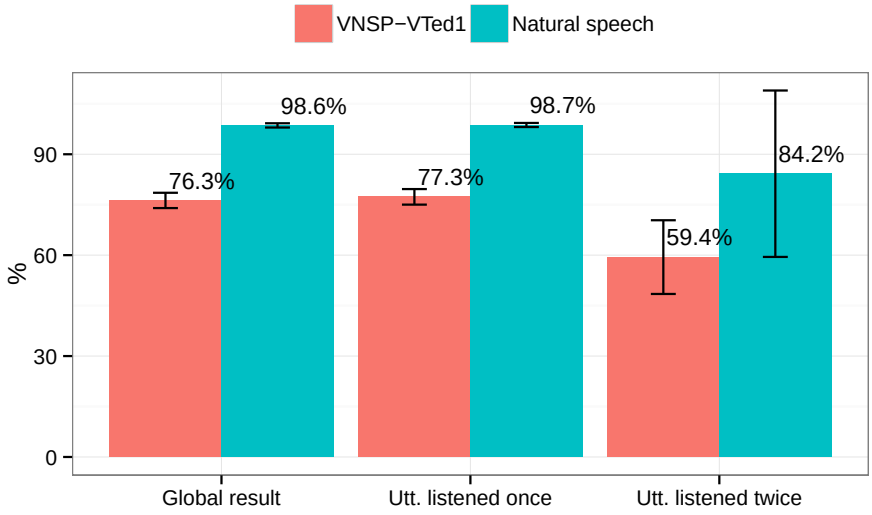


Figure 6.10 – Correct rates of tone intelligibility of initial system.

this tone was the most "difficult" for identification in context of both natural and synthetic speech.

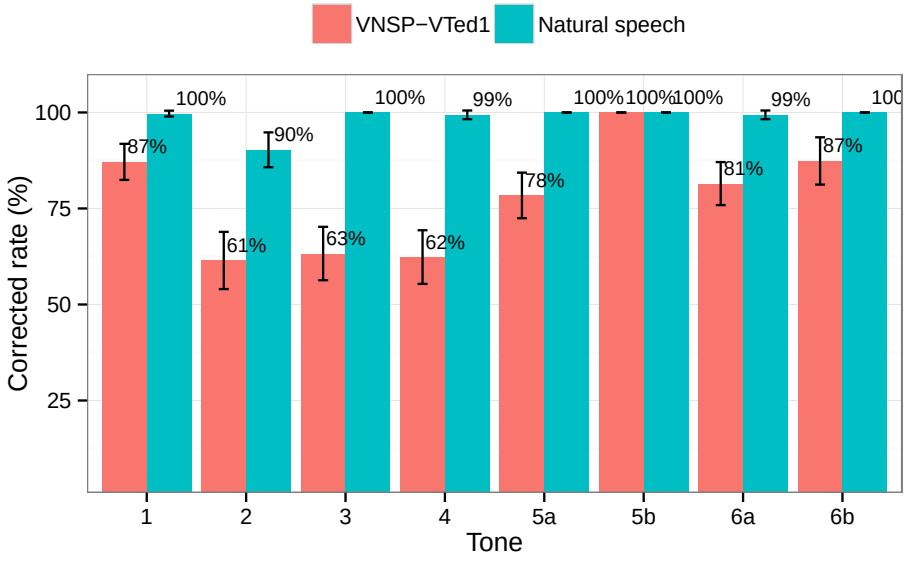


Figure 6.11 – Correct rates by tone types of tone intelligibility.

The results of "VNSP-VTed1" can be divided into three groups, according to the perceptual evaluation results. In the first group, 100% of the tones were correctly perceived, the same as the natural speech, i.e. rising tone in obstruent-final syllables (5b). The second one, with perception rates from 18 to 22% lower than natural speech, comprised the level tone 1, the drop tone 6b (around 87% correct perception) and the rising and drop tones 5a, 6a

(around 80% correct perception). The last group had perception rates from 29% to 37% lower than natural speech. This group included the broken tone 3, curve tone 4 and falling tone 2 (around 62% correct perception).

Several two-factorial ANOVAs were run on the results, illustrated in Table 6.7. The two factors were the TTS system (2 levels) and the Sentence (129 levels) or the Tone (8 levels). All factors and their interactions had highly significant effects ($p<0.05$); whereas the TTS system factor alone explained 23% (over levels of the Sentence factor) or only 12% (over levels of the Tone factor) of the variance. The Sentence factor and the interaction explained an important part of the variance, about 41% of the variance (partial $\eta^2 = 0.41$). The Tone and the interaction explained only about 4-5% each.

Table 6.6 – Anova results of the initial tone intelligibility test: first version of VTED and a natural speech

Factor	df	df error	F	p	η^2
System	1	4,386	1334.55	0.0000	0.23
Sentence	128	4,386	23.54	0.0000	0.41
System:Sentence	128	4,386	23.90	0.0000	0.41
System	1	4,628	646.25	0.0000	0.12
Tone	7	4,628	37.90	0.0000	0.05
System:Tone	7	4,628	24.43	0.0000	0.04

Tukey's Honest Significant Difference (Tukey multiple comparisons of means) with 99% family-wise confidence level were run on the results to discover the differences between the means of the levels of the "System" factor. The result showed that all means of the levels of synthetic (VNSP-VTed1) and natural speech were significantly different ($p<0.05$).

6.5.3 Final test

In this test, the option "None of the above options" was removed due to its small proportion in the initial results. The appearance of this option gave us more difficulty in data analysis. Subjects could listen to the stimuli only once. The number of stimuli presented to subjects was 390 utterances since the test corpus was designed with 130 sentences in 40 groups.

The correct rates of this experiment are illustrated in Figure 6.12. The initial version of VTED ("VNSP-VTed1") was perceived 77.4% accurately, a similar result to the initial test (76.3%). The natural speech recorded by a non-professional speaker (as the new corpus), "VDTS Natural", had almost the same correct rate (98.0%) as the one recorded by a broadcaster (as the old corpus) "VNSP Natural" (98.6%) in the initial test. This showed that this test may be considered as an "absolute" assessment of a TTS system for tonal languages, with respect to tone intelligibility. The final version of VTED, "VDTS-VTed5", received 95.4% correct rate in global result, which approached the correct rate of the natural speech (about 2.6% lower).

The correct rates by tone types are illustrated in Figure 6.13. The result of "VNSP-VTed1" had rather similar correct rates for each tone to the initial test. The correct rates of the broken tone (4), curve tone (3) and falling tone 2 were lowest, from 59% to 65%. The rising tone in sonorant-final syllables (5a) was 79% correctly identified while the three other tones: level tone (1), drop tone (6a, 6b) had quite good correct rates, from 86% to 91%. The rising tone in obstruent-final syllables was perfectly recognized (100%), the same for the final version and natural speech, as the initial test.

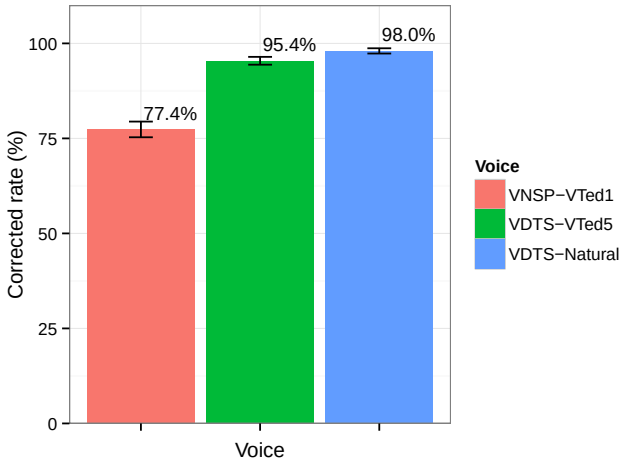


Figure 6.12 – Correct rates of the final tone intelligibility test.

All tones in the natural speech were well perceived in the same context, from 96% to 100%. The final version of VTED also received high correct rates (from 96% to 100%) for all tones, except for the falling tone (2) – only 76% correctly perceived. This result showed that the final version of VTED with the new corpus had improved considerably in terms of tone intelligibility.

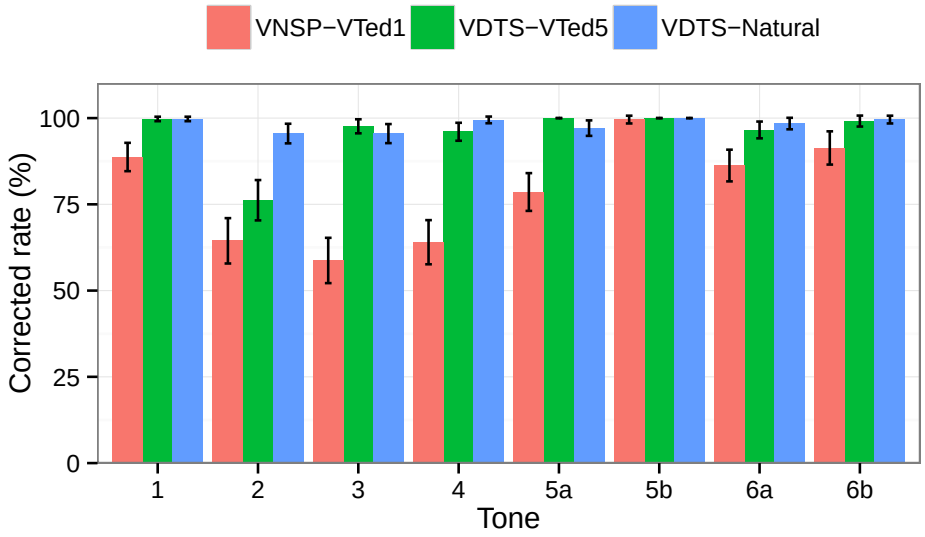


Figure 6.13 – Correct rates by tone types of final tone intelligibility test.

Several two-factorial ANOVAs were run on the results, illustrated in Table 6.7. The two factors were the TTS system (3 levels) and the Sentence (130 levels) or the Tone (8 levels). All factors and their interactions had highly significant effects ($p<0.05$). The TTS system factor alone explained only 11% (over levels of the Sentence factor) or 18% (over levels of the Tone factor) of the variance. The Sentence factor and the interaction explained an important part of the variance, about 31 to 37% of the variance (partial $\eta^2 = 0.31$ for Sentence, and

$\eta^2 = 0.37$ for the interaction). The Tone and the interaction explained only about 5% each.

Table 6.7 – Anova results of final tone intelligibility test

Factor	df	df error	F	p	η^2
System	2	7,800	842.54	0.0000	0.18
Sentence	129	7,800	27.06	0.0000	0.31
System:Sentence	258	7,800	17.68	0.0000	0.37
System	2	8,166	483.53	0.0000	0.11
Tone	7	8,166	70.64	0.0000	0.06
System:Tone	14	8,166	31.18	0.0000	0.05

Tukey's Honest Significant Difference (Tukey multiple comparisons of means) with 99% family-wise confidence level were run on the results to discover the differences between the means of the levels of the "System" factor. All TTS systems received significantly different mean scores (the p-value after adjustment for the multiple comparisons < 0.05). However, only 2.5% correct rate lower than the natural voice, the tone intelligibility of the final version of VTED (VDTS-VTed5) was approaching that of natural speech.

6.5.4 Confusion in tone intelligibility

In both the initial and final tests, we did further analysis on each lexical tone to observe the portions of other tones that subjects confused (hereafter called "confused tones"). The two tones in obstruent-final syllables (rising tone 5b and drop tone 6b) were not included in the following description as they were always in the same groups. One tone (e.g. 5b) was always confused as the other (e.g. 6b) and vice versa.

In the initial test, for the natural speech, all tones, except for the falling tone (2), were nearly 100% perceived in the same context. In the 10% error identification rate of the falling tone (2), about 9% was misheard as the curve tone (3). This may link to the fact that these two tones both have the same F0 contour shape (falling) and their voice qualities are not much different (laxness and tenseness).

Table 6.8 – Tone confusion of initial version VTED1 in the first tone intelligibility test

Tone	**F0 contour**	**Phonation**	**Initial test**		**Final test**	
			ERR	**Confusion**	**ERR**	**Confusion**
1	Level	Modal	12.87%	None	11.3%	5a (9.2%)
2	Slightly Falling	Lax	38.54%	5a, 6a (6-7%) 3 (21.0%),	35.6%	5a, 6a (6-7%) 3 (20%),
3	Falling	Tense	37.35%	2 (17.3%), 6a (12%)	41.3%	2 (21.4%), 6a (13.5%)
4	Falling-Rising	Glottal	36.73%	6a (25.6%)	36.0%	6a (26.2%)
5a	Rising	Modal	21.60%	1-4 (2-6%	21.4%	1-4 (5-6%)
6a	Dropping	Glottal	18.52%	4 (4.0%), 3 (8.3%)	13.8%	4 (4.0%), 3 (9.0%)

For the first version of synthetic speech, the tone confusion pattern, i.e. tones with considerable-confused proportions, in both (a) initial and (b) final test for tone intelligibility is shown in Table 6.8. The error rates in both tests were quite similar to each tone. The

confusion of each tone is described as follows.

- The falling tone (2) was mainly confused as the curve tone (3), about 20-21% in both tests. This pattern (2-3) was similar to the natural speech in the initial test due to the same F0 contour shape (i.e. falling).
- The curve tone (3) was distracted about 17-21% from the falling tone (2) and about 12-14% from the drop tone (6a). Despite the different falling levels, these three tones still caused substantial confusion in the subjective tests for lexical tones in the same context.
- The broken tone (4) was recognized as the drop tone (6a) in about 26% of the cases in the both tests. Two reasons for a large confusion proportion for this pattern (4-6a) might have been: (i) the same phonation type, that is "glottalization", and (ii) the same F0 contour, i.e. falling (the drop tone 6a falls more dramatically).
- The rising tone (5a) caused no clear confusion pattern in both tests. The confusion portions of this tone spread over from the tone 1 to the tone 4 (2-6% for each tone). The cause of such distraction may had come from the fact that the phonology of this tone is quite different from others. The mediocre quality of the first VTED synthetic voice might have been another reason.
- The drop tone (6a) was mostly identified about 8-9% as the curve tone (3), and 4% as the broken tone (4). The reason, as mentioned above, was that these tones have the same shape of the F0 contour (i.e. falling).

Table 6.9 – Tone confusion of last version VTED5 and natural speech in the final tone intelligibility test

Tone	F0 contour	Phonation	VTED5		Natural speech	
			ERR	Confusion	ERR	Confusion
1	Level	Modal	0.3%	None	0.3%	None
2	Slightly Falling	Lax	23.8%	3 (23.8%)	4.5%	3 (3.9%)
3	Falling	Tense	2.4%	2 (2.4%)	4.5%	6a (2.7%)
4	Falling-Rising	Glottal	4.0%	5a (3.2%)	0.5%	None
5a	Rising	Modal	0.0%	None	2.9%	4 (2.4%)
6a	Dropping	Glottal	3.4%	3 (3.2%)	1.6%	4 (1.1%)

Table 6.9 illustrates the tone confusion of the last version of VTED (VNSP-VTed5) and the natural speech in the final test for tone intelligibility. For the final version of VTED, the falling tone (2) was considered to be the most “difficult” one with the highest error rate to other tones, i.e. 24%. This result was similar to both the initial version of VTED (36-39% error rate) and the natural speech (5-10% error rate). This tone was also mostly confused with the curve tone (3), and even completely confused for the last synthetic version VTED5. Other tones in both the natural speech and VTED5 had small error rates, from 0.3-4.5%; and mainly distracted from tones having similar F0 contour shape or voice quality.

6.6 Evaluations of prosodic phrasing model

A pair-wise preference assessment was chosen for this test, which allowed us to compare two versions of VTED with different training feature sets. Subjects were asked to choose the best

speech according to them after listening to both voices.

In this test, subjects listened to 40 stimuli, composed of two utterances based on the two version needing to be compared, separated by a "beep" sound. The order of the two voices in each pair and the order of the utterances were randomly presented to the subjects. Each stimulus was presented twice to the participants.

6.6.1 Evaluations of model using syntactic rules

19 subjects (8 females) participated in the experiment. In the test corpus, 40 sentences were chosen so that each sentence covered only one syntactic rule for further analyses. Three to four examples were designed for each rule. Some sample sentences in this test can be found in Table C.4, Appendix C. This test was carried out on two synthetic voices, trained with 92% of the existing corpus VNSP: (i) "VNSP-VTed2": the initial version of VTED using the basic training feature set, and (ii) "VNSP-VTed3": VTED trained with the basic feature set plus new prosodic phrasing features using syntactic rules. In "VNSP-VTed3", sentences were automatically parsed, then **manually corrected** and passed as the input of VTED. Both voices were trained with the same corpus – VNSP.

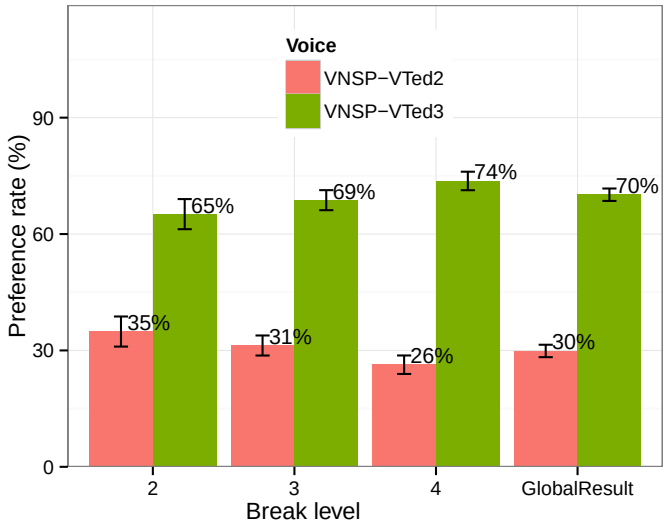


Figure 6.14 – Pair-wise comparison of VTED-VNSP with/without prosodic phrasing model using syntactic rules (manual).

The pair-comparison test results (Figure 6.14) showed a preference of about 70% for the cases of the newly proposed model over the previous one.

The MOS test was also carried out on these two versions of VTED and a natural speech reference to confirm the general quality. The MOS test results (Figure 6.15) showed an increase of 0.35 on a 5-point MOS scale, for the new prosodic-informed system (3.95/5), compared to the previous TTS system (3.61/5).

Table 6.10 shows the ANOVA results of the MOS test and the pair-wise comparison test. In the MOS test, the two-factorial ANOVA were the System (3 levels) and the Syntactic Rule (12 levels) or the Break Level (3 levels). In the pair-wise comparison test (Preference), a one-way ANOVA was run on the results to see if there was a difference between two versions of VTED, for the Syntactic Rule (12 levels) factor and the Break Level (3 levels) factor.

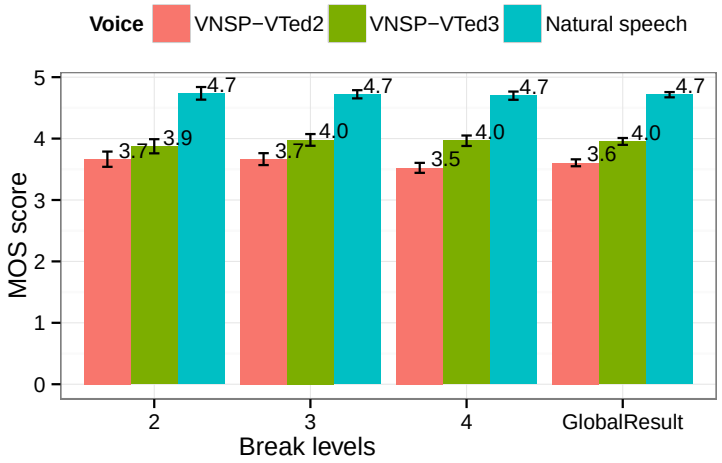


Figure 6.15 – MOS score of VTED-VNSP with/without prosodic phrasing model using syntactic rules (manual).

Table 6.10 – Anova results of MOS test and pair-wise comparison.

Test type	Factor	df	df error	F	p	η^2
MOS	System	2	2415	806.61	0.0000	0.40
	Rule	11	2415	2.98	0.0006	0.01
	System:Rule	22	2415	1.99	0.0039	0.02
MOS	System	2	2442	795.80	0.0000	0.39
	BreakLevel	2	2,442	2.66	0.0700	0.00
	System:BreakLevel	4	2442	2.72	0.0280	0.00
Preference	Rule	11	805	4.03	0.0000	0.05
	BreakLevel	2	814	2.09	0.1241	0.00

In the MOS test, the System factor had a significant effect ($p<0.05$) on the MOS score, and explained an important part of the variance (partial $\eta^2=0.40$). The Syntactic Rule factor ($p<0.05$) in both tests had a significant effect, but showed small effect strength (partial $\eta^2=0.01$ in MOS test and $\eta^2=0.05$ in Preference test). The interaction of the Syntactic Rule factor with System factor on MOS test was not significant ($p=0.0039$) and showed small effect strength (partial $\eta^2=0.02$). The effect of the Break Level factor on MOS test was not significant ($p=0.0700$) and showed almost no effect strength (partial $\eta^2=0.002$). In the pair-wise preference test, the Break Level factor was not significant, $p=0.1241$.

6.6.2 Evaluations of model using syntactic blocks

20 subjects (10 females) participated in this test. The two synthetic voices in this pair-wise preference test were: (i) VDTS-VTed4: VTED trained with the basic training feature set, and (ii) VDTS-VTed5 (final VTED): VTED trained with the basic training feature set and new prosodic phrasing features using syntactic blocks.

In "VDTS-VTed5", sentences were **automatically** parsed and passed as the input of VTED. The new corpus, VDTS, was used as a training corpus for both synthetic voices. In this test, 40 sentences with a length ranging from 2 to 26 syllables were chosen from e-

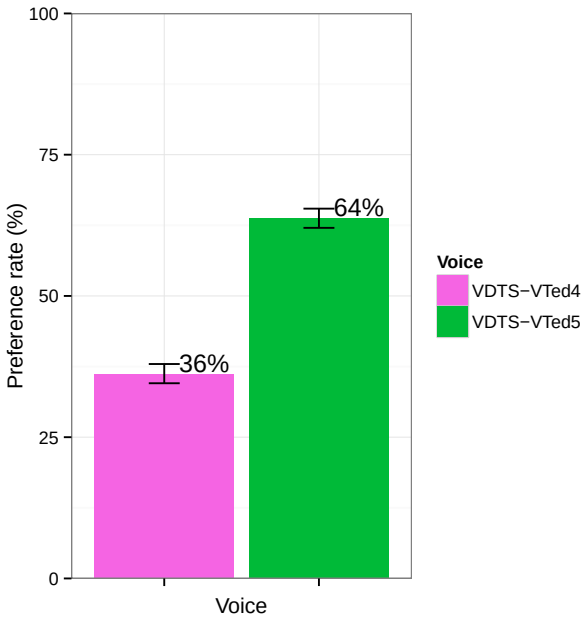


Figure 6.16 – Pair-wise comparison of VTED-VDTS with/without prosodic phrasing model using syntactic blocks (automatic).

newspapers. Some sample sentences can be found in Table C.5, Appendix C. Illustrated in Figure 6.16, about 64% of the synthetic voice with the proposed model was preferred over the previous version. This result showed that the proposed model of prosodic phrasing affected the listener perception with a positive rate.

6.7 Conclusion

Several perceptual evaluations including MOS, intelligibility, pair-wise preference, and tone intelligibility tests were conducted to assess the quality of VTED and the proposed model. Due to the special design of some tests, a test tool, **VEVA**, was developed. All perception tests took place in Vietnam.

The initial MOS test was conducted on the first version of VTED (trained with the old corpus VNSP); the previous system, HoaSung, using non-uniform unit selection with the same training corpus, and a natural speech reference. The results showed that this initial version of VTED was rather good, 0.81 (on a 5 point MOS scale) higher than HoaSung. However, the first VTED was still 1.2 point lower than natural speech. In the final MOS test, the experiment results showed that the quality of the final VTED (trained with the new corpus VDTS and the proposed prosodic phrasing model with syntactic blocks) had progressed by about 1.0 (on a 5 point MOS scale) compared to the first VTED. The proposed prosodic phrasing model was supported by VTParser, an automatic syntactic parser for Vietnamese. The gap between the synthetic speech of VTED and the natural speech was also much lessened. Although the absolute scores of the first version of VTED in the initial and final MOS tests were different (i.e. 3.6 and 2.9 on a 5 point MOS scale), the relative gaps between voices in both tests remained comparable. The first version of VTED scored about 1.2 point in the initial test, and about 1.5 point in the final test, lower than the natural speech. The natural voice in the final test was about 0.3 point lower than the one in the initial test. It turned out that

participants in the final test gave "stricter" and more "sensitive" rates than in the first one.

Perceptive testing showed that with new prosodic phrasing models, the synthetic speech of VTED was preferred about 64% (automatic parsing and syntactic blocks) or about 70% (manual parsing and syntactic rules) over the previous version. Both subjective and objective evaluations were performed to confirm that ToBI features did not ameliorate the quality of Vietnamese TTS systems in general, and even degraded in some cases. As a result, ToBI features were removed from the second version of VTED.

In the tone intelligibility test, groups of meaningful sentences with the same syllables and the same syllable order, diverging by only one tone, were prepared. Subjects were asked to choose the most likely syllable they had heard among a group of syllables bearing different tones in an utterance. In the initial test, about 23% on average and – depending on the tone type – from 0% to 37% difference from the natural speech were perceived. In the final test, the last version of VTED received high correct-rates, from 96% to 100%, for all tones except for the falling tone 2 – only 76% correctly perceived. The global correct-rate of the final VTED in identifying tones in context was only 2.6% lower than that of the natural speech. The falling tone 2 was identified as the most "difficult" with the highest error rates, i.e. 4% (final test) or 10% (initial test) for the natural speech, and about 24% (final test) or 39% (initial test) for the synthetic speech. The gaps between the first VTED and the natural speech in the both tests remained similar, i.e. about 22% in the initial test, and 21% in the final test. We believed that this kind of perceptual evaluation could be considered as an "absolute" assessment since the subjects did not have to give "score", but only had to choose the most likely syllable they had heard. The tone confusion pattern, i.e. tones with considerable confused proportion, was found for the first VTED. The falling tone (2) and the curve tone (3) was mainly distracted from each other. The cause of such distraction might have come from the same F0 contour shape (i.e. falling) of these two tones. The broken tone (4) was mostly confused as the drop tone (6a). The two reasons for this confusion might have been: (i) the same phonation type, that is "glottalization", and (ii) the same F0 contour, i.e. falling (the drop tone 6a falls more dramatically).

The intelligibility test was designed with Latin square matrix 3x3 for three voices: the first VTED, the final VTED and a natural speech. At the syllable-level, the error rate of the first VTED was about 14.3%, hence about 12.0% higher than that of the natural speech. This first version diverged by about 5.8-7.5% from the natural speech at the lower levels, i.e. tone and phoneme. The gap between the last VTED and the natural speech was only from 0.4% - 1.4%. This result showed that the last version of VTED considerably advanced in terms of intelligibility to the first one, and that the intelligibility of this version approached that of the natural speech.

Chapter 7

Conclusions and perspectives

Contents

7.1 Contributions and conclusions

The main contributions of this work can be recapitulated as follows: (i) studying background and adopting technique for the research, (ii) proposing a new speech unit – tonophone (iii) designing, recording, and pre-processing a new corpus, VDTS, which covered both phonemic and tonal contexts, for the Vietnamese TTS, (iv) proposing a prosodic phrasing model using automatic syntactic information for the Vietnamese TTS, (v) designing, building a complete Vietnamese HMM-based TTS system – VTED, and (vi) evaluating the TTS system with several perception tests.

7.1.1 Adopting technique and performing literature reviews

Different areas of knowledge were studied for this research.

Adopting speech synthesis technique. Speech synthesis techniques were investigated for the comparison and analysis. Statistical parameter and unit selection speech synthesis are the two prominent state-of-the-art methods. While statistical parameter speech synthesis can be simply described as generating the average of some sets of similarly sounding speech segments, the purpose of unit-selection synthesis is to retain natural unmodified speech units using parametric models that offer other benefits. Both those techniques mainly depend on data. In the concatenative approach, the data is effectively memorized, whereas in the statistical approach, the general properties of the data are learned. While many possible approaches to statistical synthesis are possible, most research has focused on using hidden Markov models (HMMs). The initial motivation of this work was to build a high-quality TTS system to assist Vietnamese blind people in accessing written text. Hence, the HMM-based approach was chosen for this Vietnamese TTS system due to its predominance on general quality, footprint and robustness. This technique was then studied from its statistical approach using machine learning to its implementation.

Studying Vietnamese phonetics and phonology. A new domain of knowledge, the phonetics and phonology of the Vietnamese language, was discovered. A number of new terminologies as well as phonetic realizations of the language were studied from various phonetic books and articles. All phonetics and phonology information necessary to build a TTS system were considered. These included the language characteristics, the syllable structure, the phonological system, and the lexical tone system.

Although Vietnamese is an alphabetic script, there exist a number of characteristics that distinguish it from occidental languages. Vietnamese is a tonal language, in which pitch is mostly used as a part of speech, changing the meaning of a word/syllable. Moreover, it is an inflectionless language in which its word-forms never change; an isolating language, the most extreme case of an analytic language where the boundaries between syllables and between morphemes are the same, and each morpheme is a single syllable. Vietnamese is a quite fixed order language with the general word order SVO (subject-verb-object). Blanks are not only used to separate words, but also used to separate syllables that make up words. In addition, many of Vietnamese syllables are not only words by themselves, but can also be part of multi-syllable words whose syllables are separated by blanks.

Vietnamese syllables have a hierarchical structure. Its first layer is comprised of two main parts: an initial consonant and a rhyme. A tone is a non-linear or suprasegmental part of a syllable, and mainly adheres to the rhyme. Tones appear simultaneously with segmental

elements of rhyme, i.e. medial, nucleus and ending. The nucleus and tone are compulsory while others are optional.

There are 19 initial consonants in the modern Hanoi, which is considered the standard Vietnamese. Hanoi Vietnamese allows eight segments in final position, distinguishes nine long vowels, four short vowels, and three falling diphthongs, which have the same function as vowels in the syllable.

The Vietnamese tone system belongs to the pitch-plus-voice quality type, i.e. the lexical tone is not defined solely in terms of pitch: it is a complex bundle of pitch contour and voice quality characteristics. Although there are six lexical tones in the writing system, Vietnamese has a six-tone paradigm for sonorant-final syllables: the level tone 1, the falling tone 2, the curve tone 3, the broken tone 4, the rising tone 5a, and the drop tone 6a, and a two-tone paradigm for obstruent-final syllables: the rising tone 5b and the drop tone 6b. The broken tone 4 and the drop tone 6a in sonorant-final syllables are glottalized: the tone 4 has medial glottal constriction and ends on a high F0 value; the tone 6a has glottal constriction throughout, and is typically falling. The tones 5b and 6b (i.e. the tones of syllables ending in /p/, /t/ or /k/) are not glottalized, either in final or non-final position.

Studying Vietnamese syntactic parsing. The syntax theory and Vietnamese syntactic parsing were examined for discovering the relationship between syntax and prosodic phrasing. Studying syntax helps us understand the sentence construction, and the possibilities of arrangements of the elements in sentences. Grammatical categories include not only the Part Of Speech (POS), e.g. noun, verb, preposition but also types of phrase, e.g. noun phrase, verb phrase, prepositional phrase. Two major aspects of sentence syntactic structures are: (i) phrase structure grammar (or constituency structure grammar) concerning the organization of the units that constitute sentences, e.g. Sentence → Prepositional phrase + Noun phrase + Verb phrase; and (ii) dependency grammar (or relational structure) concerning the function of elements in a sentence such as subject, predicate or object.

In automatic syntactic parsing, to resolve ambiguities, a treebank, a (usually large) collection of sentences, is used by supervised learning methods. Each sentence in the treebank is annotated with a complete syntactic parsing by human expert. The generative models, i.e. choosing possible derivations–a sequence of decisions, give lower accurate than the discriminative ones, i.e. modeling ambiguities using arbitrary features of parse trees. Perceptron, a discriminative model, is the most-widely-used technique due to its simplicity and performance. The adopted parsing model; which uses an averaged perceptron–a global linear model, and A* search to guarantee the optimality; can balance both accuracy and parsing speed. On the same test set of English Treebank, the experiment results showed a high and balance performance of the adopted model over other state-of-the-art models (F-score=91.1%, speed=13.6 sentences/s).

VTParser was the Vietnamese syntactic parser for the TTS system. It was trained with VietTreebank and used the adopted model for constituency parsing with adaption for Vietnamese language. This parser had the best performance (F-score=81.6%, speed=13.6 sentences/s) over other state-of-the-art Vietnamese syntactic parsers on the same test corpus. Due to different approaches of prosodic phrasing modeling for our TTS system, three types of syntactic parsing were proposed and implemented in VTParser: (i) The standard constituency parsing: normal constituency parsing, (ii) The unnamed constituency parsing: all constituent elements above words are named the same as "XP"; and (iii) The constituency parsing with functional labels: standard constituency parsing with functional labeling using averaged perceptron algorithm. The standard constituent parsing without phrase names has

the highest precision (84.43%) and F-score (85.40%) while the quality of the constituent parsing with functional labels is lower by about 14%. There is a small gap between the accuracy of the standard constituent parsing and the unnamed one (about 3% difference).

7.1.2 Proposing a new speech unit – tonophone

Based on the literature study, due to the great importance of lexical tones, a "tonophone" – an allophone in tonal context — was proposed as a new speech unit for our work. To build the tonophone set of the system, the lexical tone was taken into account and adhered to all allophones in the rhyme, and the initial consonant maintained its form without any information of the tone. As a result, a tonophone set with 207 tonophones was constructed from 48 Vietnamese allophones. This unit set includes: (i) 19 initial consonants without tone information, (ii) medial and 16 nucleus adhering to eight tones, (iii) unreleased final stops adhering to two tones 5b, 6b, and (iv) other final consonants adhering to six tones 1-4, 5a, 6a. An acoustic-phonetic tonophone set of Vietnamese was also built for (i) HMM clustering using phonetic decision trees, and (ii) automatic labeling, i.e. automatic segmenting and force-aligning the speech corpus with the orthographic transcriptions. Based on the literature review, main phonetic attributes were specified for both consonants and vowels on this acoustic-phonetic unit set, such as place or articulation or manner of articulation for consonants, position of tongue or height for vowels.

Grapheme-to-phoneme rules were developed for transcribing Vietnamese consonants and vowels/diphthongs. Many graphemes can be directly converted to tonophones without any ambiguity, such as "b-" to [ɓ], "ch-, tr-" to [tɕ], "-m" to [m], "ê" to [e]. Well-defined rules were found for more complicated cases/variants. For instance, for the grapheme "a", if it is followed by "nh" or "ch", the phoneme is [ɛ̆]; if it is followed by "u" or "y", the phoneme is [ă]; otherwise, the phoneme is [a]. The full G2P rules were used for both transcribing the raw text for corpus design and building the G2P conversion module of our TTS system.

PRO-SYLDIC, a Vietnamese syllable pronunciation e-dictionary was constructed for filtering pronounceable syllables in text normalization as well as transcribing texts. Pairs of syllable orthography and transcription in the dictionary were automatically generated mainly based on (i) the G2P rules, (ii) syllable orthographic rules, and (iii) the list of rhymes. A table of 170 Vietnamese rhymes with not only existent but also pronounceable ones in the language was designed. The reason to maintain all pronounceable rhymes is that the input of a Vietnamese TTS system may include numerous loanwords that includes nonexistent but pronounceable syllables, as well as newly appeared words from teenagers or Internet users. The PRO-SYLDIC was constructed by combining 19 initial consonants with 772 rhymes bearing tones, making a total of 21,648 Vietnamese pronounceable syllables (orthography).

7.1.3 Designing and building a new corpus

A new corpus was always a big concern during the development process of a TTS system. A method for building a phonetically-rich and -balanced corpus in terms of both phonemic and tonal contexts from a big raw text was presented. The following task sequence was applied on designing new corpora: (i) Raw text collection, (ii) Raw text pre-processing, (iii) Corpus design, (iv) Corpus recording, and (v) Corpus pre-processing. Although most of these tasks were programmed for automatically performing, a lot of manual tasks still had to be done. The reason was that there were many exceptional cases, as well as human errors.

Building a huge and phonetically-rich and -balanced raw text. Since the raw text was considered to be represented for the language in terms of phonetic distribution in corpus design, it was huge and collected from various sources (i.e. e-newspapers, e-stories, examples from the Vietnamese e-dictionary, existing resources, and special design). There was a need to pre-process to make it to be suitable for the design process with five main tasks: (i) Sentence segmentation, (ii) Tokenization, (iii) Text cleaning, (iv) Text normalization, and (v) Text transcription. Texts were first segmented into sentences, and then tokenized into syllables or Non-Standard Words (NSWs – which cannot be directly transcribed to phonemes, e.g. numbers, dates, abbreviations, currency). Sentences having more than 70 syllables or containing unreadable characters (e.g. control ones) were removed. The next task for "cleaned" sentences were was text normalization, in which NSWs were then processed and expanded to speakable syllables. The normalized text was finally transcribed into tonophones to provide a suitable input for further process. The raw text included 323,934 clean sentences, and more than 10 billions syllables.

Since Vietnamese is a tonal language, our work targeted to design a phonetically-rich and -balanced corpus in both phonemic and tonal context. Hence, two proposed speech units used for designing corpora for Vietnamese TTS systems were: (i) a "tonophone": a phone regarding the lexical tone of the bearing syllable, and (ii) a "di-tonophone": an adjacent pair of "tonophones". The di-tonophone set was constructed using a dictionary including meaningful syllables (theoretical), and using the raw text (real). The phonetic distribution of the raw text, which was considered as a reference for the corpus design, was calculated for different speech units, including the two new ones.

Designing corpora. The whole corpus design included a number of iterations of selection process, whose output was the best candidate in terms of phonetic-richness and -balance at the current state of the uncovered units and their distributions. The selection process could be described as follows. A subset of the huge raw data including the rarest/most frequent uncovered unit [1] was extracted to achieve a set of candidate sentences. Due to the simplicity and effectiveness, the greedy algorithm was adopted to search for the best candidate sentence (with the highest weight) among that subset. The weight of a sentence was the proportion of its uncovered distinct unit number and its total distinct unit number. After each selection, the uncovered unit set was updated, and the selection process was repeated until a constraint (e.g. coverage, condition) was satisfied.

To examine the performance of the proposed design process, we carried out the design that considered di-tonophones as speech units with a constraint of the target corpus size. The output was a new text corpus, "SAME", with a similar size (i.e 24,164 number of phones) as the old one – VNSP. The bounded size of the target corpus was small and the number of di-tonophones appearing once in the raw data was considerable. If sentences containing the rarest speech unit were considered first, sentences including those single-occurence di-tonophones would have been the unique candidates and hence have been chosen for the target corpus. Therefore, sentences containing the *most frequent speech unit* were chosen as candidates for weight calculation for maximizing the coverage of the target corpus. The results show that with a similar syllable number, the coverage of the new corpus "SAME" was much higher than the old one. There was no or small difference of the phone or tonophone

1. We assumed that in the phonetically-rich and -balanced raw text, sentences containing rare speech units might include more common ones, but the possibility of vice versa was much smaller. Therefore, the rarest uncovered speech units should be processed first. However, if the coverage of the target unit is not 100% (e.g. 70%), the most frequent unit may be chosen to optimize the corpus size. The corpus design may need to be performed twice to find out the best solution for choosing the rarest or most frequent uncovered speech unit.

coverages, yet wide gaps (about 17-22%) of the other unit coverages between these two corpora. The di-tonophone coverage of the new corpus reaches 52.4%, while that of the old one was only 29.6%. The VSYL corpus with a complete syllable coverage was also designed for improving the quality of the non-uniformed unit selection speech synthesis.

The target training corpus for our TTS system, the VDTS (Vietnamese DiTonophone Speech) corpus, was designed with a full coverage of di-tonophones since it is necessary to record all the transitions between any two tonophones. Obviously, the VDTS corpus had 100% coverage of phones, tonophones, and di-phones. Its coverages of initial/rhymes and syllables were 95.1% and 70.2% respectively.

Recording and pre-processing speech corpus. A total of 5,338 sentences including the VDTS and VNSP corpora, and some other sentences (called VDTO – Vietnamese Di-Tonophone and Others) for the evaluation phase were recorded by a female non-professional native speaker from Hanoi, aged 31 (named Thu-Trang). The recordings were conducted in a well-equipped studio including a soundproof vocal booth and a control station at the LIMSI-CNRS Laboratory, Orsay, France. There were 27 one-hour recording sessions to produce nearly 8 speech hours. The speech quality was controlled during the sessions by a supervisor. After several recording sessions, audio files were checked to ensure the global quality and to reduce errors for next sessions.

Utterances in the speech corpus were renamed by sentence codes and pre-processed for our Vietnamese TTS system. They were automatically segmented and force-aligned to build an annotated corpus by the EHMM labeler. However, there existed wrongly-labeled breath noises, which made discontinuous transitions between syllables of the preliminary synthetic voice. These breath noises were hence semi-automatically corrected for a final annotated corpus.

The VDTS speech corpus that was used for training VTED finally contains 3,947 utterances in about 6.4 hours (384 minutes). The speech rate of VDTS was about 9.6 phonemes/s, or 3.6 syllables/s, hence about 25% lower than the previous one recorded by a broadcaster. In average, the speaker Thu-Trang produced a perceived pause every nine syllables, about 16% more pauses than the broadcaster.

7.1.4 Proposing a prosodic phrasing model

In the HMM-based speech synthesis, prosodic cues such as F0 or duration can be well learned in both phonemic and tonal context. The remaining problem in prosodic analysis is prosodic phrasing, the process of inserting prosodic breaks in an utterance. It includes pause insertion and lower levels of grouping syllables. In an HMM-based TTS system, a pause is considered a phoneme; hence its duration can be modeled. However, the appearance of pauses cannot be predicted by HMMs. Lower phrasing levels above words may not be completely modeled with basic features. To the best of our knowledge, there is no such work on the Vietnamese language. Due to the constraint with the lexical tones, the utterance-level intonation in Vietnamese language may be less important in prosodic phrasing than that in other intonational languages (e.g. English, French). In this research, we aimed at prosodic phrasing for the Vietnamese TTS using durational clues alone for two levels of prosodic phrasing for Vietnamese TTS: (i) pause appearance – one of the most prominent and frequent levels, and (ii) final lengthening.

In a preliminary study, syntactic rules between constituent or dependent elements were proposed to predict three break levels after an iterative refinement process. Some statistical treatments of pause length and final lengthening at predicted boundaries were done for refin-

ing rules in a small corpus VNSP (630 sentences with manual annotations). These syntactic rules could work well, i.e. P=91.2% and F-score=69.7%, in the manual environment but only constituent syntactic rules gave an acceptable results (i.e. P=84.2%, F-score=39.9%) in the automatic environment. Furthermore, with an automatic syntax parser, predicted break levels did not differ significantly leading to only one break level, i.e. pause appearance.

Another approach, which was finally implemented to the final version of VTED, was proposed to use syntactic blocks for predicting final lengthening and pause appearance. Syntactic blocks were proposed as syntactic phrases with a bounded number of syllables. The analysis corpus was the VDTO corpus including 5,338 utterances in about 7.7 hours. Audio files in this corpus were automatically segmented at phoneme-level by EHMM labeler. Phonemes were then grouped to syllables and perceived pauses in a different tier. Text files were automatically parsed to syntax trees by the VTParser, the adopted Vietnamese syntactic parser using shift-reduce parsing with averaged perceptron.

A normalized syllable duration (ZScore) pattern of syntactic blocks was found to be similar to that of breath groups (containing syllables between two consecutive perceived pauses): (i) slightly shortening at the first syllable, (ii) shortening at the penultimate one, and (ii) strong degree of lengthening at the last one. Final lengthening even still existed but with a lower degree in the last syllables of syntactic blocks excluding last syllables of higher level ancestors and breath groups. As a result, two levels of prosodic phrasing using durational clues alone were identified: (i) pause appearance using syntactic blocks with a maximum of 10 syllables and (ii) final lengthening using syntactic blocks with a maximum of 6 syllables.

Improvements were done by some strategies of grouping single syntactic blocks for final lengthening. The pause appearance prediction was improved by combining the syntactic blocks with two additional predictors: (i) syntactic link, a syntactic-tree-based relationship/distance between two grammatical words; and (ii) POS. Some rules were constructed for predicting pause appearance. The performance of the rule-based model with the three predictors was good, i.e. P=84.8% and F-score=65.8% at the analysis phase; P=84.9% and F-score=67.4% at the testing phase.

A predictive model was experimented in a 10-fold cross validation with these three predictors using J48 (i.e. the Java implementation of the C4.5 algorithm, a decision tree approach for the classification problem). The "Syntactic block" was the most important predictor since the model with only this predictor had the best Precision (83.4%) and Recall (71.1%), compared to models with only POS (F-score=43.6%) or syntactic link (F-score=52.6%) alone. The "Syntactic link" predictor helped the model improve the Recall (6% improved) while "POS" gave effective information to increase the Precision (4% improved). The complete model including the three predictors had the best results with Precision=89.0%, Recall=74.6%, hence F-score=81.2%. Using a separate test set (VDTO-Testing), the performances of the two models were slightly different. The Precision of the predictive model was a bit higher than the rule-based one (i.e. nearly 3%). However, the recall of the predictive model considerably improved, was about 14% higher than the rule-based one. This model could be automatically built for any speaker or any dialect by machine learning techniques, and hence was chosen for applying to the final version of VTED.

New prosodic training features were proposed for Vietnamese HMM-based TTS systems. The perceptual evaluations showed that the voice trained with the new proposed prosodic features was preferred about 64%-70% to the one without these new features.

7.1.5 Designing and constructing VTED

Since the initial motivation was to build a high-quality TTS system for the blind, it should be complete. Much effort was put on designing and constructing VTED, a complete Vietnamese HMM-based TTS system. The architecture of VTED composed of three parts: (i) Natural language processing (NLP), (ii) Training, and (iii) Synthesis. From the input text, the NLP part extracts contextual features to provide for both Training and Synthesis phase. In the Training phase, these factors were aligned with speech unit labels and trained with speech parameters (i.e. spectral and excitation) to build context dependent HMMs. In the Synthesis phase, according to a label sequence with these factors, contextual features were used to produce a sequence of speech parameters. Finally, a synthetic speech was obtained using these speech parameters and a vocoder.

Contextual features for Vietnamese were chosen at tonophone, syllable, word, phrase and utterance levels inheriting from other works with adaption for Vietnamese. In a stable structure with composed of four elements, although only nucleus is mandatory, the appearance of other elements is also common. To capture the context of tonophones between elements inside a syllable as well as the transitions to previous/next syllables, two preceding and two succeeding tonophones were chosen for the phonemic context of the current tonophone. About 84% Vietnamese words are compound words, of which 70% have two syllables, only 1% have more than four syllables. Therefore, only one preceding and one succeeding syllable or higher levels were considered as features for their contexts. There were locative factors of the current tonophone, syllable, word or phrase in the current syllable, word, phrase or utterance. The numbers of two lower levels in the 3-gram model were also considered, e.g. at Word level: the number of tonophones in the previous/current/next word, and the number of syllables in the previous/current/next word. We had features on functions of phonemes in syllable structures (onset or coda), punctuations and some prosodic features (e.g. POS of previous/current/next word, break index of phonemes, syllable position type). The lexical tone of the previous/current/next syllable was an important feature in Vietnamese.

After carefully studying different platforms, Mary TTS was chosen for building VTED due to its ease of use, expandability, and a high quality vocoder. It is an open-source, multilingual Text-to-Speech Synthesis platform and with a big development community in GitHub. This platform facilitates building a TTS system for a new language with a well-designed process.

A lot of works were done on building a separate natural language processing part for VTED. Although some modules were adopted from other works, they consumed not only effort but also time for integrating (POS tagger, syntax parser), adapting and extending (Word segmentation), or even re-developing (Text Normalization). Much time was also spent for building dictionaries, e.g. the transcribed syllable dictionary PRO-SYLDIC and MEA-SYLDIC, the loan word dictionary, the abbreviation dictionary. The two modules from scratch were the G2P Conversion and the Prosody Modeling. On the other hand, most of the existing Mary TTS modules had to be adapted and modified for tonal languages, especially for a big unit number of the tonophone set.

7.1.6 Evaluating the TTS system

Several perceptual evaluations including MOS, intelligibility, pair-wise preference, and tone intelligibility tests were conducted to assess the quality of VTED and the proposed model. Due to the special design of some tests, a test tool, **VEVA**, was developed. This was a portable tool, which could be deployed in any operating system since all specific-platform aspects were considered during the design and implementation. The output data of this

tool was stored in XML files with a well-designed structure, facilitating its extensibility and portability. This test tool was deployed and used in both Windows and Mac OS X for the above tests. The Graphical User Interface (GUI) of the tests in VEVA was designed with a full-screen mode, which prevented subjects from getting distracted by other objects (e.g. icons, applications, and bars) on the screen.

All perception tests took place in Vietnam. Some statistical treatments were done on the test results to confirm the reliability of the tests. The perceptual results showed that the last version of VTED was approaching the natural speech, in terms of the general and tone intelligibility. The general naturalness of the last version considerably progressed over the first implementation, and had a small gap to the natural speech.

Different perception tests concerning tonal aspects. The initial MOS test was conducted on the first version of VTED (trained with the old corpus VNSP); the previous system, HoaSung, using non-uniform unit selection with the same training corpus, and a natural speech reference. The results showed that this initial version of VTED was rather good, 0.81 (on a 5 point MOS scale) higher than HoaSung. However, the first VTED was still 1.2 point lower than natural speech. In the final MOS test, the experiment results showed that the quality of the final VTED (trained with the new corpus VDTS and the proposed prosodic phrasing model with syntactic blocks) had progressed by about 1.0 (on a 5 point MOS scale) compared to the first VTED. The proposed prosodic phrasing model was supported by VTParser, an automatic syntactic parser for Vietnamese. The gap between the synthetic speech of VTED and the natural speech was also much lessened. Although the absolute scores of the first version of VTED in the initial and final MOS tests were different (i.e. 3.6 and 2.9 on a 5 point MOS scale), the relative gaps between voices in both tests remained comparable. The first version of VTED scored about 1.2 point in the initial test, and about 1.5 point in the final test, lower than the natural speech. The natural voice in the final test was about 0.3 point lower than the one in the initial test. It turned out that participants in the final test gave "stricter" and more "sensitive" rates than in the first one.

Perceptive testing showed that with new prosodic phrasing models, the synthetic speech of VTED was preferred about 64% (automatic parsing and syntactic blocks) or about 70% (manual parsing and syntactic rules) over the previous version. Both subjective and objective evaluations were performed to confirm that ToBI features did not ameliorate the quality of Vietnamese TTS systems in general, and even degraded in some cases. As a result, ToBI features were removed from the second version of VTED.

The intelligibility test was designed with Latin square matrix 3x3 for three voices: the first VTED, the final VTED, and a natural speech. At the syllable-level, the error rate of the first VTED was about 14.3%, hence about 12.0% higher than that of the natural speech. This first version diverged by about 5.8-7.5% from the natural speech at the lower levels, i.e. tone and phoneme. The gap between the last VTED and the natural speech was only from 0.4% - 1.4%. This result showed that the last version of VTED considerably advanced in terms of intelligibility to the first one, and that the intelligibility of this version approached that of the natural speech.

In the tone intelligibility test, groups of meaningful sentences with the same syllables and the same syllable order, diverging by only one tone, were prepared. Subjects were asked to choose the most likely syllable they had heard among a group of syllables bearing different tones in an utterance. In the initial test, about 23% on average and – depending on the tone type – from 0% to 37% difference from the natural speech were perceived. In the final test, the last version of VTED received high correct-rates, from 96% to 100%, for all tones except

for the falling tone 2 – only 76% correctly perceived. The global correct-rate of the final VTED in identifying tones in context was only 2.6% lower than that of the natural speech. The gaps between the first VTED and the natural speech in the both tests remained similar, i.e. about 22% in the initial test, and 21% in the final test. We believed that this kind of perceptual evaluation could be considered as an "absolute" assessment since the subjects did not have to give "score", but only had to choose the most likely syllable they had heard.

Tone confusion in the same tonal context. The falling tone 2 was identified as the most "difficult" with the highest error rates, i.e. 4-10% for the natural speech, and about 24%-39% for the synthetic speech. As aforesaid, the last version of VTED received high correct-rates, from 96% to 100%, for all tones except for the falling tone 2 – only 76% correctly perceived.

We found a tone confusion pattern, i.e. tones with considerable confused proportion, for the first VTED as follows.

- The falling tone (2) was mainly confused as the curve tone (3), about 20-21%. This pattern (2-3), which is also similar to the natural speech, might have been due to the same F0 contour shape (i.e. falling) of these two tones.
- The curve tone (3) was distracted about 17-21% from the falling tone (2) and about 12-14% from the drop tone (6a). Despite the different falling levels, these three tones still had substantial confusion rates in these subjective tests for lexical tones in the same context.
- The broken tone (4) was recognized as the drop tone (6a) about 26%. The two reasons for this large confusion for the pattern (4-6a) might have been: (i) the same phonation type, that is "glottalization", and (ii) the same F0 contour, i.e. falling (the drop tone 6a falls more dramatically).
- The rising tone (5a) had no clear confusion pattern. The confusion portions of this tone spread over the tone 1 to the tone 4 (2-6% for each tone). The cause of such distraction might have come from the fact that the phonology of this tone was quite different from others. The mediocre quality of the first VTED synthetic voice might have been another reason.
- The drop tone (6a) is mostly identified about 8-9% as the curve tone (3), and 4% as the broken tone (4). The reason, as mentioned above, was that these tones have the same shape of F0 contour (i.e. falling).

7.2 Perspectives

This research resulted in several achievements towards building a high-quality TTS system for Vietnamese. However, further work can be done to ameliorate the quality of the system, as well as to expand the research to a range of applications. Some major perspectives of this work are presented here.

7.2.1 Improvement of synthetic voice quality

The synthetic voice quality of the last VTED was good, and even reached the natural speech in terms of general and tone intelligibility. However, it was still distinguishable from the natural speech in the MOS test (about 0.5 point lower than natural speech on a 5 point

scale). In the tone intelligibility, although most of the tones received high correct-rates (from 96% to 100%) the same as the natural speech, the falling tone 2 was wrongly identified about 24%. With a small corpus, the error rate of the synthetic voice was 23% on average and ranged from 0% to 37% depending on the tone type. This result showed that the lexical tones were not well modeled for specific cases, even with a good corpus.

The tone correctness of the synthetic speech may be improved by investigating the structures of the decision tree in the tree-based context clustering process of the HMM-based training, such as the work for Thai: Chomphan (2011), Chomphan and Chompunth (2012), Chomphan and Kobayashi (2007, 2008),Moungsri et al. (2014). The work of Chomphan and Kobayashi (2007) proposed the use of tone groups and tone types for designing four different structures of decision tree including a single binary tree structure, a simple tone-separated tree structure, a constancy-based-tone-separated tree structure, and a trend-based-tone-separated tree structure.

On the other hand, a recording corpus with electroglottograph (hereafter EGG) signals may help in ameliorating the tone intelligibility and intonation of the synthetic speech. The EGG signal is a "reliable measurement of the surface of vocal fold contact" (Michaud, 2004). Therefore, with the EGG signals, the F0 can be calculated in a more accurate and reliable way than other traditional algorithms (e.g. ESPS algorithm using the snack tool in this work). Since F0 is one crucial speech parameter (i.e. excitation parameter) in HMM-based training, we believe that the more accurate F0 was extracted, the better quality of speech is generated.

Zen et al. (2013) highlighted three major factors that degrade the quality of the HMM-based synthesized speech: vocoding, accuracy of acoustic models, and over-smoothing. This work addressed the accuracy of acoustic models by a new prominent learning technique, i.e. Deep Neural Networks (DNN). The authors reported that the use of the DNN for the decision tree can address some limitations of the conventional approach such as the inefficiency in expressing complex context dependencies, fragmenting the training data, and completely ignoring linguistic input features that did not appear in the decision trees. The experimental results of this work showed that the DNN-based systems outperformed the HMM-based systems with similar number of parameters. The work of Wu et al. (2015) showed that the hidden representation used within a DNN could be improved through the use of Multi-Task Learning, a simple and convenient way to provide additional supervision during training. The experimental results confirmed the effectiveness of the proposed methods, and the stacked bottleneck features in particular offer a significant improvement over both a baseline DNN and a benchmark HMM system.

7.2.2 TTS for other Vietnamese dialects

Although the modern Hanoi Vietnamese is considered the standard Vietnamese, there is still a need to extend VTED to be able to synthesize other popular Vietnamese dialects for local residents. Three other main dialects are Hue, Nghe An (Center of Vietnam), and Southern Vietnam.

The two main tasks to develop a TTS system for a Vietnamese dialect include: (i) studying the phonetics and phonology of that dialect, (ii) designing and recording a new corpus for it. These tasks can reuse the results of the current work and make an adaption for the target dialect. The new corpus then can be automatically segmented, labeled and trained for a new voice of that dialect.

7.2.3 Expressive speech synthesis

Expressive speech synthesis has been considered to accelerate the quality of TTS systems to a new vision, which will help users accept TTS outputs by producing less impersonal speech. For example, a TTS system can make a voice sound happy or subdued, friendly or empathic, authoritative or uncertain. There exist several research directions on this topic such as (i) Speaker characterization and voice personalization: models that can be adapted to a speaker thus taking into account their mood, personality or origins, (ii) Duration and prosody modeling: Prosody control (e.g. phoneme duration, melody control) to have a voice adapted to the context of interaction.

With expressive speech synthesis, we can automatically generate high-quality audiobooks in which various reading styles can be adapted for different characters. Different voices can be used for generating dialogues for different characters. Emotional synthetic speech can capture the multitude of emotional expressions (e.g. “anger”, “fear”, “joy” and “sad”) of each character in dialogues. Similarly, text subtitles in movies can be also automatically converted to expressive speech with little effort.

7.2.4 Voice reader

The Sao Mai Vietnamese reader is a concatenative synthesis system with a corpus whose syllables were isolately recorded. In spite of the low quality and other drawbacks, the Sao Mai Vietnamese reader currently has been considered the most common software supporting the Vietnamese blind to use personal computers. Although there are several other Vietnamese readers targeting blind users, they have not been used in real-life due to their usability limitations.

According to a survey in the work of Tran (2013), there is no better Vietnamese reader targeting the blind users than the Sao Mai voice. However, users really need to have a better quality reader for improving their interactions with computers. A high-quality TTS system can also facilitate the new users in learning how to use computers in a less time-consuming way.

From the design and implementation of VTED, there is further work necessary to construct such a reader. First, some optimizations in designing corpora as well as contextual feature set should be conducted. Second, the interactions between the TTS system and its environment need to be improved. Finally, the necessary functions of a reader have to be investigated and developed based on the user requirements.

7.2.5 Reading machine

Another perspective of TTS is embedded systems, which are specialized machines for reading speech from other formats, such as text, image. It can be even combined with a scanner and an OCR application to provide users a data entry from paper data records.

For the Vietnamese language, this machine will be very useful not only for the blind and the people with low vision, but also for any user who wants to listen to e-newspapers, journal, or even messages without reading. Based on the current TTS system for Vietnamese, a lightweight TTS system needs to be designed and developed on embedded systems. “Flite+hts_engine” [2] can be used as a good reference for such a system.

2. http://hts-engine.sourceforge.net/

List of publications

1. **Nguyen Thi Thu Trang**, Rilliard Albert, Tran Do Dat, and d'Alessandro Christophe. Prosodic phrasing modeling for vietnamese TTS using syntactic information. *In 15th Annual Conference of the International Speech Communication Association (Interspeech 2014)*, pages 2332–2336. Singapore, September 2014a. ISCA.

2. **Nguyen Thi Thu Trang**, Tran Do Dat, Rilliard Albert, Alessandro Christophe, and Pham Thi Ngoc Yen. Intonation issues in HMM-based speech synthesis for Vietnamese. *In The 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU'14)*, pages 98-104. St. Petersburg, Russia, May 2014b.

3. **Nguyen Thi Thu Trang**, Alessandro Christophe, Rilliard Albert, and Tran Do Dat. HMM-based TTS for Hanoi Vietnamese: issues in design and evaluation. *In 14th Annual Conference of the International Speech Communication Association (Interspeech 2013)*, pages 2311-2315. Lyon, France, August 2013b. ISCA.

4. **Nguyen Thi Thu Trang**, Pham Thanh Thi, and Tran Do-Dat. A method for Vietnamese text normalization to improve the quality of speech synthesis. *In Proceedings of the First Symposium on Information and Communication Technology (SoICT'10)*, pages 78–85, Hanoi, Vietnam, 2010. ACM. ISBN 978-1-4503-0105-3.

5. Do Van Thao, Tran Do Dat, and **Nguyen Thi Thu Trang**. Non-uniform unit selection in Vietnamese speech synthesis. *In Proceedings of the Second Symposium on Information and Communication Technology (SoICT'11)*, pages 165–171. Hanoi, Vietnam, 2011. ISBN 978-1-4503-0880-9.

6. Le Anh-Tu, Tran Do-Dat, and **Nguyen Thi Thu Trang**. A Model of F0 Contour for Vietnamese Questions, Applied in Speech Synthesis. *In Proceedings of the Second Symposium on Information and Communication Technology (SoICT'11)*, pages 172–178. Hanoi, Vietnam, 2011. ACM. ISBN 978-1-4503-0880-9.

Appendix A

Vietnamese syntax parsing

Contents

A.1 Syntax theory

This section gives an introduction to the syntax theory, i.e. the study of phrase and sentence structures, which is a base for the Vietnamese syntax.

A.1.1 Syntax and grammar

Syntax. A term from ancient greek, Syntax's literal meaning is "arrangement" or "setting out together". It refers to "the branch of grammar dealing with the ways in which words, with or without appropriate inflections, are arranged to show connections of meaning within the sentence" (Valin, 2001, p. 1).

Example 9 *Modifier*–modified and *possessor*–possessed in English and Vietnamese

- *Modifier*–modified:
  - In English: "*blue* tables"; in Vietnamse: "những chiếc bàn *màu xanh*"
  - In English: speak French *fluently*; in Vietnamse: "nói tiếng Pháp *trôi chảy*"
- *Possessor*–possessed:
  - In English: *my* house; in Vietnamse: "ngôi nhà *của tôi*"
  - In English: Ted's *kittens*; in Vietnamse: "những chú mèo con *của Ted*"

Studying syntax tells us how to construct sentences, and a number of "possible arrangements of the elements in sentences". The order of the main elements in a sentence is one of the most noticeable and crucial ways in which languages differ (Valin, 2001). For instance, in both English and Vietnamese, the subject comes before the verb and the object follows the verb. In a sentence like "He *put* a pen on the table" in English, the order of subject-*verb*-object is the same as in Vietnamese: "Anh ta *đặt* một chiếc bút trên bàn". However, the order of elements in the relationships modifier–*modified*, possessor–*possessed* is reverse for these two languages, illustrated in Example 9. In English, the modifier elements ("blue") of nouns always precede, e.g. "*blue* tables" while the modified nouns ("những chiếc bàn") in Vietnamese are always followed by modifiers ("màu xanh"), e.g. "những chiếc bàn *màu xanh*". The modifiers (e.g. "fluently" in English, "trôi chảy" in Vietnamese) come after the modified verbs (e.g. "speak" in English, "nói" in Vietnamese) in both Vietnamese and English (sometimes inverse in English). Nevertheless, the order of possessor and possessed elements in English (i.e. possessors precedes processed) are reverse to those in Vietnamese (i.e. possessors come after processed).

It appears that syntax is a "central component" of human language, which has often been "characterized as a systematic correlation between certain types of gestures and meaning", illustrated in Figure A.1. "Changes in the word form to indicate their function in the sentence" are referred to as "*inflections*", and "the study of the word formation and how they may change their form" is called "*morphology*" (Valin, 2001). English is an inflection language, which has morphological markers of grammatical cases, numbers, and tenses; whereas, Vietnamese is an inflectionless language whose word forms never change and it has no finite/non finite distinction. In other words, Vietnamese words do not change depending on grammatical categories, e.g. "bạn" is the same for singular and plural (while "student" and "students" in

English), the verb "**học**" ("study" in English) never changes in any tense: there are tense markers, for example "*đã*" ("already" in English) for the past tense: "*đã* **học**" ("studied" in English), "*đang*" ("-ing" in English) for continuous tense: "*đang* **học**" ("studying" in English).

Figure A.1 – Language as a correlation between gestures and meaning (Valin, 2001, p. 3).

Grammar. The term "*morphosyntax*", explicitly showing the important relationship between syntax and morphology, is alternatively known as "*grammar*", composing syntax and morphology (Valin, 2001, p. 2). Grammar is often used to refer to "the complete set of rules needed to produce all the regular patterns in a given language" or roughly means "all the structural properties of the language except sound structure (phonology) i.e. the structure of words, phrases, sentences, texts, etc". It can help us analyze and describe the word and sentence patterns of a language by formulating a set of rules with respect to those patterns (Kroeger, 2005, p. 5).

Grammatical categories and syntactic structures. To classify words into "*grammatical categories*" is a natural first step toward allowing grammars to capture generalizations. The term "*grammatical category*" now covers not only the Parts Of Speech (POS), e.g. nouns, verbs, prepositions but also types of phrase, e.g. noun phrases, verb phrases, prepositional phrases. Parts of speech are termed as "*lexical categories*" in contemporary linguistics or traditionally referred as "*word classes*", whereas "*non-lexical categories*" or "*phrasal categories*" means types of phrase (Valin, 2001)(Kroeger, 2005). For convenience, they will be abbreviated such as "V" to refer "verb", "VP" to refer "verb phrase".

In the syntactic structure of sentences, "two distinct yet interrelated aspects" must be distinguished: (i) Phrase structure grammar concerns the organization of the units that constitute sentences, e.g. Sentence → Prepositional phrase + Noun phrase + Verb phrase, and (ii) Dependency grammar encompasses the dependency relation, e.g. subject–predicate. A detail of the different grammatical categories and syntactic structures will be covered in the next subsections.

A.1.2 Parts Of Speech (POS)

As mentioned, a part of speech are also called a word class or a lexical category. The most important lexical categories are nouns (N), verbs (V), adjectives (A), adverb (R) and prepositions (E). In traditional grammar, the lexical categories are defined using notional (i.e. semantic) properties. For instance, a noun is a word that names "a person, place or thing", verb is defined as an "action word" or "event word", and adjective is defined as "a word expressing a property, attribute or state". However, these characterizations fail to identify nouns like "destruction", "theft", "beauty", "heaviness". They cannot distinguish between the verb "love" and the adjective "fond (of)", or between the noun "fool" and the adjective "foolish" (Kroeger, 2005). As result, in modern linguistics, parts of speech are "defined morphosyntactically in terms of their grammatical properties", which are (i) its position in the sentence and (ii) its morphology (Valin, 2001)(Kroeger, 2005). For instance in English, the word "fool" belongs to the word class that can be modified by adjectives and inflected for number, have no comparative form, and can occur as subjects. Hence the word "fool" is

classified to a "noun" while the word "foolish" is an adjective, due to their positions (e.g. subject) and forms (e.g. inflected, comparative) in the sentence.

Nouns can be categorized in numerous ways, e.g. proper nouns (Np, i.e. proper name), common nouns (i.e. not refer to unique individuals or entities). Common nouns (N) may be divided into mass nouns and count nouns. Count nouns, as the name implies, denote countable entities while mass nouns are not readily countable in their primary senses. Pronouns (P) are closely related to nouns, and "traditionally characterized as substitutes for nouns or as standing for nouns". Verbs (V) may be classified along various dimensions, hence "quite complex and is more appropriately in the domain of semantics rather than syntax". For instance, intransitive verbs take just a subject, transitive verbs take a subject and a object, ditransitive verbs take a subject, object and object. Another dimension concerns the kind of situation it represents: "static situations", "a change of state" or "complex situations involving an action plus a change of state" (Valin, 2001, p. 6).

The parts of speech in a particular language can be assigned a label based on universal notional patterns. Each language has its own lexical categories, therefore a deep study on those for English is not presented here. A description of Vietnamese word classes will be presented later.

A.1.3 Phrase structure grammar

A sentence does not consist simply of a string of words; and not the case that "each word is equally related to the words adjacent to it in the string" (Valin, 2001). Words in a sentence may be grouped into grammatical units of various sizes. One crucial unit is the clause, "the smallest grammatical unit which can express a complete proposition". A sentence may consist of just one clause or several clauses. A single clause may contain several phrases, another important unit. A single phrase may contain several words, which may contain several morphemes. "Each well-formed grammatical unit (e.g. a sentence) is made up of constituents which are themselves well-formed grammatical units", such as clauses, phrases, etc. There are only a limited number of basic types of units, which is adequate for a large number of languages: sentence, clause, phrase, word, morpheme. This kind of structural organization is called a part–whole hierarchy: each unit is entirely composed of smaller units (Kroeger, 2005, p. 32-33).

Terminologies in hierarchical structure. Some terminologies of tree structure, which may be used later in this work, are presented as follows. The tree elements are called "nodes". The lines connecting elements are called "branches". Nodes without children are called leaf nodes, "end-nodes", or "leaves". Every finite tree structure has a member that has no superior. This member is called the "root" or root node. The root is the starting node. A node's parent is a node one step higher in the hierarchy (i.e. closer to the root node) and lying on the same branch. A node's child is a node one step lower in the hierarchy (i.e. further to the root node) and lying on the same branch. Sibling ("brother" or "sister") nodes share the same parent node. A node's "uncles" are siblings of that node's parent. A node's "nephews" are siblings of that node's child. A node that is connected to all lower-level nodes is called an "ancestor". The connected lower-level nodes are "descendants" of the ancestor node.

Lowest common ancestor (LCA) of two nodes a and b in a tree is the lowest (i.e. deepest) node that has both nodes as descendants, where we define each node to be a descendant of itself. If the node a has a direct connection from the node b, the node b is the lowest common ancestor and vice versa.

Phrases and phrasal category. The term "phrase" in linguistics has a more precise meaning other than "any group of words". That is a group of words that function as a

constituent (i.e. a unit for purposes of word order) within a simple clause. Phrases may be classified into different categories, such as noun phrases, verb phrases.

> Two phrases belong to the same category if they have the same grammatical properties. Two basic types of evidence are useful for determining whether two phrases belong to the same category. These are: (i) sameness of distribution (i.e. mutual substitutability); and (ii) sameness of internal structure. The criterion of mutual substitutability involves the general principle that two phrases of the same category could potentially occur in the same positions, unless one of them is inappropriate for semantic reasons. For example, phrases that can occur in subject or object position are generally noun phrases. In identifying word classes, "internal structure" means morphological structure, for example the capacity to be inflected for number (in the case of nouns) or tense (in the case of verbs). When we are dealing with phrases, "internal structure" means the category and order of the phrase's constituents. For example, an English noun phrase frequently begins with a determiner (a, the, this, that). The criterion of mutual substitutability (sameness of distribution) involves the general principle that two phrases of the same category could potentially occur in the same positions, unless one of them is inappropriate for semantic reasons. For example, phrases which can occur in subject or object position are generally noun phrases.
>
> –Main idea mostly extracted from Kroeger (2005, p. 35-36)–

Phrase heads. There exists one word in most phrases being the most important element of the phrase, called the *head* (H) of the phrase. The category of phrase heads in general gives name to the phrase. For instance, the phrase "those beautiful places" is a noun phrase as its head word, i.e. "places", is a noun, while the phrase "considerably important" is an adjective phrase, since its head word, i.e. "important", is an adjective. The remaining concern here is how to know which element in the phrase is the head and how to distinguish the head from its dependents (i.e. all the other elements in the phrase). There are three following specific ways in which the head is more "important" than the other elements (Kroeger, 2005, p. 36-37).

> First, the head of a phrase determines many of the grammatical features of the phrase as a whole, e.g. the head noun determines grammatical number for the subject noun phrase as a whole ("the new kittens *are* in the barn"). And second, the head may determine the number and type of other elements in the phrase. For example, we take the verb to be the head of a clause, and different verbs require different numbers and categories of phrases to occur with them in their clause. Dependents which are selected by the head word in this way are referred to as complements. Thus subjects, objects, etc. are often referred to as complements of the verb. To take another example, many adjective phrases contain a prepositional phrase complement as the choice of preposition is determined by the identity of the head adjective, e.g. "John felt [sorry *for his actions*]". Finally, the head is more likely to be obligatory than the modifiers or other non-head elements. For instance, all of the elements of those noun phrases: "pigs", "the pigs", "the three pigs", "the little three pigs" can be omitted except the head word "pigs". If this word is deleted, the result is ungrammatical.
>
> As noted above, the head of a phrase will generally be a lexical item of the same category – a noun phrase will be headed by a noun, an adjective phrase by an

> adjective, etc. However, not all lexical categories can be heads of phrases. Those that can (including at least Noun, Verb, Adjective, and Preposition in English) are called major categories; those that cannot (e.g. conjunctions) are called minor categories. Major categories are typically open classes; these categories contain an indefinite but large number of words, and new words tend to be added frequently through borrowing or innovation. Minor categories are typically closed classes; such classes contain only a small, fixed number of words, and new words are added very slowly. However, this correlation is not perfect, e.g. preposition is a major category but probably a closed class.
>
> –Main idea mostly extracted from Kroeger (2005, p. 36-37)–

Coordinate and subordinate sentences. There are two basic ways in which one clause can be embedded within another: coordination vs. subordination. In a coordinate structure, two constituents belonging to the same category are conjoined to form another one of that category. In a coordinate sentence, two (or more) main clauses (or independent clauses – S) occur as daughters and co-heads of a higher clause. A dependent clause (or subordinate clause – SBAR, i.e. complement clauses, adjunct or adverbial clauses and relative clauses) is one that functions as a dependent, rather than a co-head. This combination of words cannot stand-alone or form a complete sentence, but provides additional information to finish the thought (Kroeger, 2005).

Phrase structure rules. Sentences are constructed using a set of rules (their "internal grammar"), that could produce the sentence structures and, ultimately, all other possible sentence patterns in the language. These rules are known as phrase structure rules (PS rules) and have the following form: A → B C, for example **PP** → E *NP* (a **prepositional phrase** composes of a preposition and a *noun phrase*, e.g. "on *the tree*").

The following example illustrates this hierarchical structure in Vietnamese: $[_S\ [_{NP}\ [_N$ Cô giáo (The teacher)$]$ $[_{Np}$ tiếng Anh (English)$]$ $[_{SBAR}$ mà (who) $[_{NP}\ [_N$ anh (you)$]]$ $[_{VP}$ đã $[_V$ gặp (met)$]]$ $[_{NP}\ [_N$ hôm qua (yesterday)$]]$ $_{SBAR}]$ $_{NP}]$ $[_{VP}$ đang $[_V$ đọc$]$ (is reading) $[_{NP}\ [_N$ sách (books)$]]$ $[_{PP}\ [_P$ trong (in)$]$ $[_{NP}\ [_N$ thư viện (the library)$]_{NP}]_{PP}]$ $_{VP}]$ $_S]$.

A.1.4 Dependency structure grammar

The phrase structure grammar by its rules cannot provide an adequate account of what speakers say. For instance, some sentences produced from phrase structure rules like "Mary sings a white cake" or "John likes" are incoherent in English: they are either "semantically ill-formed" or "ungrammatical". Another type of complication that can arise is structural ambiguity, i.e. the sentence as a whole is ambiguous because it has more than one possible phrase structures, even though none of the individual words is ambiguous in this context.

To (partly) address those issues, another important aspect of sentence structure need to be considered, namely "*grammatical relations*". Those are the *syntactic function* of elements such as subjects or objects in a sentence. Therefore, this type of syntax is referred to as "*relational structure*". This is also termed as "*dependency structure*" since it actually encompasses the dependency relation.

Aside from the predicate itself, the elements of a simple clause, i.e. clausal dependents, can be classified as either adjuncts or arguments, illustrated in Figure A.2. Adjuncts (ADT) are elements that are "not closely related to the meaning of the predicate but which are important to help the hearer understand the flow of the story, the time or place of an event, the way in which an action was done, etc". Adjuncts can be omitted without creating any

sense of incompleteness. Arguments are those elements that are "selected by the verb"; they are "required or permitted by certain predicates, but not by others". In order to be expressed grammatically, arguments must be assigned a grammatical relation within the clause. There are two basic classes of grammatical relations: obliques (or indirect arguments) vs. terms (or direct arguments). Terms (i.e. subject–SUB, primary object–OBJ, secondary object–OBJ2) "play an active role in a wide variety of syntactic constructions", while obliques (OBL) are "relatively inert" (Kroeger, 2005, p. 62).

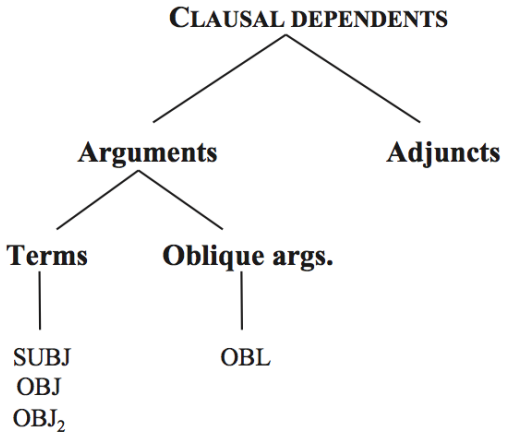


Figure A.2 – Classification of clausal elements (Kroeger, 2005, p. 62).

Subject and object. Subject (SUB) of a sentence is traditional defined as "the doer of the action" or "what the sentence is about", while the object is the person or thing "acted upon by the doer". However, these definitions seem to work for many sentences but fails in others, such as "John was bitten by a dog" or "As for Bill, I wouldn't take his promises very seriously". Rather, grammatical criteria must be used to develop a suitable definition, in which subjects have "some grammatical properties that other elements of the sentence do not share" (Kroeger, 2005, p. 56). For instance, in English, the subject normally comes before the verb, while the object and other parts of the sentence follow the verb. Other properties can be found in Kroeger (2005, 56-57).

Predicate. In traditional grammars, the term "predicate" of a sentence or clause is frequently used to refer to everything in a clause that is not part of the subject, telling us what the subject does or is. However, in this work, three kinds of predicate are considered: (i) a simple predicate is an element which identifies the property or relationship, usually verbs in English (ii) a compound predicate consists of two or more simple predicates (iii) a complete predicate (PRD) composes of a simple predicate and all accompanying modifiers and other words receiving the action or completing its meaning. For instance, in a sentence in English like "Her sons are eating a big cake", the simple predicate is "are eating", while the full predicate is "are eating a big cake". The term "predicate" alone is to refer to a simple predicate. Predicates may describe a situation which is changing over time (i.e. events) or a situation which is relatively static – unchanging (i.e. states).

Argument: term and oblique. It appears that three general syntactic functions are arguments, modifiers and predicates (Valin, 2001). For instance, in the above sentence, "her sons" and "a big cake" are the arguments; "are eating" is the predicate; and "her", "a" and "interesting" are the modifiers. The arguments are the participants which "must be involved because of the very nature of the relation or activity named by the predicate, and without which the clause cannot express a complete thought". For example, any event named by

the predicate "eat" must involve at least two participants, the eater and the eaten. It turns out that the predicate "eat" takes two arguments. A clause then can be also referred as a "grammatical unit which expresses a single predicate and its arguments". As a result, arguments show a closer and significant grammatical relationship between their predicate and its subject or object than that between the predicate and other elements of the clause (Kroeger, 2005).

In order to be expressed grammatically, arguments must be assigned a grammatical relation within the clause: obliques (or indirect arguments) vs. terms (or direct arguments). Terms includes subject, primary object (i.e. the object agrees and is closest to the verb) and secondary object. For example, in the sentence "John gave *Mary* his old radio", "Mary" is the primary object while "his old radio" is the secondary object. In English, all oblique arguments are marked with prepositions, whereas subjects and objects are expressed by "bare noun phrases". For instance, in a sentence like "*Michael Jackson* donated *his sunglasses* to the National Museum", terms are "Michael Jackson" (subject) and "his sunglasses" (object), oblique argument is "to the National Museum" (Kroeger, 2005).

Adjunct. Besides arguments relating closely to the meaning of the predicate, other elements of meaning need to be conveyed as well. Those elements are important to help the hearer understand the flow of the story, the time or place of an event, the way in which an action was done, etc. They are called *adjuncts*. It is not always easy to distinguish adjuncts from oblique arguments. It turns out that adjuncts are never obligatory, since they are not implied by (or directly related to) the meaning of the verb. In other words, adjuncts are *always* deletable, whereas arguments *may* be optional. For instance, in a sentence like "George *intentionally* put the money into his pocket *last night*", *adjuncts* are "intentionally" (showing manner) and "last night" (showing time) while arguments are "the money" and "into his pocket".

Some clausal dependents are illustrated in the following example for Vietnamese: $[_S[_{ADT}$ Tối qua (last night)] $[_{SUB}$ Kiên (Kien)] $[_{PRD}$ đã tặng (gave) $[_{OBJ}$ một bó hoa hồng (a bouquet of roses)] $[_{OBL}$ cho mẹ của anh ấy (to his mother)] $_{PRD}]$ $_S]$.

A.2 Syntactic parsing techniques

Parsing or syntactic analysis in general is the process of analyzing a string of symbols, either in natural language or in computer languages, complying with the rules of a formal grammar. In natural language processing, the syntactic analysis (hereafter called syntactic parsing) may vary from low to high levels. The lowest level can be referred as simply part-of-speech tagging for each word in the sentence. Shallow parsing (also known as "chunking", "light parsing") decomposes of sentence structure into constituents but not specify their internal structure nor their role in the main sentence. The highest level parsing, i.e. the full parsing, can recover not only the phrase structure of a sentence, but also can identify the sentence structure dependency between each predicate in the sentence and its explicit and implicit arguments. In syntactic parsing, ambiguity is a particularly onerous issue since the most probable analysis has to be chosen from an exponentially large number of alternative analyses. As a result, parsing algorithms plays an important role to handle such ambiguity, hence decides the quality of a parser corresponding to different levels from tagging to full parsing.

This section describes parsing models with the use of supervised machine learning as well as how to design features to deal with ambiguity.

A.2.1 Treebank corpus

A classical method to recover syntactic structure and relation is to design a grammar of the language, that is a set of syntactic rules. A context-free grammar (CFG) [1] is a particular type of formal system to compose syntactic rules. It has proved very useful in the precise characterization of computer languages and also serves as the starting point for much work in syntactic theory.

However, designing out a CFG to cover all syntactic analysis of natural language is problematic. Unlike programming language, natural language is too complex to simply list all the syntactic rules in terms of a CFG. This grammar can be extended to include more syntactic constructions, however listing all possible syntactic structures in a language has a number of issues. In addition, it is also problematic to exhaustively list lexical properties of words, e.g. all grammar rules in which a particular word can be a participant. This is known as a typical *knowledge acquisition* problem.

On the other hand, there is another less apparent problem: the explosion of rule combinations. It turns out that the rules could combine with each other in explosive ways, leading to a extremely large and ambiguity space of many redundant incorrect syntactic constructions. Consider a simple CFG that provides a syntactic analysis of noun phrases as a binary branching tree: Rules: $N \rightarrow NN$. Recursive rules produce ambiguity: with N as the start symbol, for the input "natural language processing", two candidate parses are:

[$_N$ [$_N$ natural] [$_N$ language]] [$_N$ processing]; [$_N$ natural] [[$_N$ language] [$_N$ processing]]

This is a second knowledge acquisition problem: It it necessary to know not only the set of syntactic rules for a particular language, but also which analysis is the most plausible for a given input sentence – called *ambiguity* problem.

In order to address those problems, a common way is to construct a *treebank*, which is a data driven approach to syntactic analysis. A treebank is simply a collection of sentences (normally a large sample of sentences, also called a corpus of text), where each sentence is provided by a complete syntactic analysis. The syntactic analysis for each sentence should have been annotated by human expert to guarantee the most plausible analysis for that sentence. Before the annotation process, an *annotation guidelines* is typically written in order to ensure a consistent scheme of annotation throughout the treebank.

Treebank solves the first knowledge acquisition problem by finding the grammar underlying the syntax analysis. Obviously, there is no set of syntactic rules or linguistic grammar, as well as there is no list of syntactic constructions provided explicitly in a treebank. In fact, the parser can infer a set of implicit grammar rules to cover a large amount of syntactic analysis that does not exist in treebank. Concerning the second problem, since each sentence in a treebank has been given its most plausible syntactic analysis, some supervised learning methods can be used to train a scoring function over all possible syntactic analyses of that sentence. For a given sentence that is not seen in the training data, a statistical parser can use this scoring function to return the syntax analysis that has the highest score, which is taken to be the most plausible analysis for that sentence.

A.2.2 Generative models

The purpose of generative models is to resolve the above ambiguity problem of syntactic parsing. The main idea of the generative models could be simply described as follows: In

1. CFGs consist of an initial symbol [q.v.], a finite lexicon with words classified into grammatical categories [q.v.], and a finite collection of rules of the form $A \rightarrow \omega$, where A is a single symbol (representing a type of phrase), and ω is a finite string of lexical and/or phrasal categories (Sag et al., 2003).

order to find the most plausible parse tree, the parser has to choose between the possible derivations, each of which can be represented as a sequence of decisions. Let each derivation $D = d_1, ..., d_n$ is the sequence of decisions used to build the parse tree. Then for an input sentence x, the output parse tree y is defined by the sequence of steps in the derivation. A probability for each derivation is introduced as in Equation A.1.

$$P(x, y) = P(d_1, ..., d_n) = \prod_{i=1}^{n} P(d_i | d_1, ..., d_{i-1}) \tag{A.1}$$

$$P(d_1, ..., d_n) = \prod_{i=1}^{n} P(d_i | \Phi(d_1, ..., d_{i-1})) \tag{A.2}$$

$$= \prod_{i=1}^{n} P(d_i | \phi_1(H_i), ..., \phi_k(H_i)) \tag{A.3}$$

where

- $P(x, y)$: Probability of the derivation to build the output parse tree y from the input x
- $d_1, ..., d_n$: Sequence of steps in the derivation
- d_i: Step i in the derivation.

The conditioning context in the probability $P(d_i | d_1, ...d_{i-1})$ is called the *history* and corresponds to a partially built parse tree. A simplifying assumption can be made to keep the conditioning context to a finite set by grouping histories into equivalence classes using a function Φ, illustrated in Equation A.2. Using Φ, each history $H_i = d_1, ..., d_{i-1}$ for all x, y is mapped to a fixed finite set of feature functions of the history $\phi_1(H_i), ..., \phi_k(H_i)$. In terms of these k feature functions, the probability of each derivation can be calculated as Equation A.3.

Probabilistic Context-free Grammars (PCFG). Probabilistic Context-free Grammars (PCFG) model is the simplest classical instance of generative models, where the parse tree y having the highest joint probability $p(x, y)$ with the input sentence x. In this model, scores or probabilities are assigned for each rule in the CFG grammar in order to provide a score or probability for each derivation (or a partial parse tree). The probability of a derivation is simply the sum of scores or product of probabilities of all CFG rules used in that tree. In order to make sure that the probability of the set of trees generated by a PCFG is well-defined, the probability of each rule $N \rightarrow \alpha$ is calculated as Equation A.4. It appears that such probability distribution can be easily computed from the Treebank. For for simplicity, the chart-based algorithm then can be used to find the candidate parse tree with highest probability as a final result.

$$P(N \rightarrow \alpha) = \frac{count(N \rightarrow \alpha)}{count(N)} \tag{A.4}$$

Lexical PCFG. The most popular and classical generative model is Lexical Probabilistic Context-free Grammars (LPCFG) of Collins (1999). Its idea is to extend the history of a parse tree by adding more information of phrase head words. On the test set, that is the section 23 of English Penn Treebank (Marcus et al., 1993), the LPCFG parser can reach $Fscore$ of 88.2%, while $Fscore$ of the parser with the naive PCFG is 73%.

Latent Variables PCFG. The current state-of-the-art generative parsing model in terms of accuracy belongs to the well-known Berkeley parser (Petrov and Klein, 2007). These authors assumed that if it is possible to split each constituent label (even POS) in Treebank in a good manner, a high accuracy can be obtained. This method is called *"Latent Variables PCFG"*, which uses the Expectation-Maximization (EM) algorithm to find the best manner to split their grammar, reaching $Fscore$ of 90.1%. In syntactic parsing, Berkeley parser has been considered as one of the strongest one because it does not need any grammar information, only the Treebank corpus, making it easily apply into any languages.

A.2.3 Discriminative models

The definition of PCFG means that various rule probabilities had to be adjusted in order to obtain the right scoring of parses. Meanwhile, the independence assumptions in PCFG, which are dictated by the underlying CFG, often leads to bad models. These models cannot use information vital to the decision of rule scores leading to high scoring plausible parses. Such ambiguities can be modeled using arbitrary "features" of the parse tree. Discriminative methods provide us with such a class of models. Even in common machine learning, the performance of the discriminative models is usually better than that of the generative models.

Collins (2002) created a simple framework that described various discriminative approaches to train a parsing system (and also chunking or tagging). This framework was called a *global linear model* (Collins, 2002). Let X be a set of inputs, and Y be a set of possible outputs parse trees. A global linear model could be described as follows:

- Each $x \in X$ and $y \in Y$ is mapped to a d-dimensional feature vector $\Phi(x, y)$, with each dimension being a real number, summarizing partial information contained in (x,y).
- The function $GEN(x)$ generates the set of possible outputs y for a given x.
- A weight parameter vector $\mathbf{w} \in R^d$ assigns a weight to each feature in $\Phi(x, y)$, representing the importance of that feature. The value of $\Phi(x, y) \cdot w$ is the score of pair (x,y). The higher the score is, the more plausible it is that y is the output for x.

Having $\Phi(x, y)$, $\mathbf{w}$, and $GEN(x)$ specified, the highest scoring candidate y^* from $GEN(x)$ should be chosen as the most plausible output, as shown in Equation A.5, where $F(x)$ returns the highest scoring output y^* from $GEN(x)$.

$$F(x) = \arg \max_{y \in GEN(x)} p(y|x, \mathbf{w}) \tag{A.5}$$

$$\log p(x|x, \mathbf{w}) = \Phi(x, y) \cdot \mathbf{w} - \log \sum_{y' \in GEN(x)} \exp(\Phi(x, y') \cdot \mathbf{w}) \tag{A.6}$$

$$F(x) = \arg \max_{y \in GEN(x)} \Phi(x, y) \cdot \mathbf{w} \tag{A.7}$$

where

- $GEN(x)$: a set of possible outputs y for a given x,
- $\mathbf{w}$: a weight parameter vector,
- $\Phi(x, y)$: a d-dimensional feature vector.

Inputs: Training Data $\langle (x_1, y_1), ..., (x_m, y_m) \rangle$; number of iterations T
Initialization: Set $\mathbf{w} = 0$;
Algorithm:
  for $t = 1 \dots T$ **do**
    for $i = 1 \dots m$ **do**
      $y'_i = \text{argmax}_{y \in GEN(x)} \Phi(x_i, y) \cdot \mathbf{w}$
      if $(y'_i \neq y_i)$ **then**
        $\mathbf{w} = \mathbf{w} + \Phi(x_i, y_i) - \Phi(x_i, y'_i)$
      end if
    end for
  end for
Output: The updated weight parameter vector $\mathbf{w}$.

Figure A.3 – The original perceptron learning algorithm.

Commonly, a *conditional random field* (Lafferty et al., 2001) could be used to define the conditional probability as a linear score for each candidate y and a *global* normalization term as Equation A.6. However, a simpler global linear model, shown in Equation A.7, can be obtained by ignoring the normalization term (thus much faster to train). Many experimental results in parsing have shown that this simpler model often provides the same or even better accuracy than the more expensively trained normalized models.

A well-known discriminative model in syntactic parsing will be presented in the next subsection, that is **Perceptron**. This approach was adopted for the Vietnamese syntactic parsing, a module in our high-level speech synthesis.

A.2.4 Perceptron

A perceptron (Rosenblatt, 1988) originally introduced as a single-layered neural network. In structured prediction problem such as parsing, perceptron could be considered as the most widely-used model due to its simplicity and efficiency. Comparing to the generative model or other discriminative models, it is much simpler while still keeping a competitive accuracy (Carreras et al., 2008, Collins and Roark, 2004, Zhu et al., 2013). Perceptron could be trained by using the online learning, that is, processing examples one at a time, during which it adjusts a weight parameter vector that can then be applied on input data to produce the corresponding output. The weight adjustment process awards features appearing in the truth and penalizes features not contained in the truth. After the update, the perceptron ensures that the current weight parameter vector is able to correctly classify the present training example.

Original perceptron. Suppose we have m examples in the training set. The original perceptron learning algorithm is shown in Figure A.3. The weight parameter vector $\mathbf{w}$ is initialized to $\mathbf{0}$. Then the algorithm iterates through those m training examples. For each example x, it generates a set of candidates $GEN(x)$, and picks the most plausible candidate, which has the highest score according to the current $\mathbf{w}$. After that, the algorithm compares the selected candidate with the real one (from Treebank), and if they are different from each other, $\mathbf{w}$ is updated by increasing the weight values for features appearing in this top candidate. If the training data is linearly separable, meaning that it can be discriminated by a function that is a linear combination of features, the learning has been proven to converge in a finite number of iteration (Collins, 2002).

This original perceptron learning algorithm is simple to understand and to analyze. How-

Training Phase
Inputs: Training Data $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$; number of iterations T
Initialization: Set $\mathbf{w} = 0$;
Algorithm:
for $t = 1 \ldots T$ **do**
for $i = 1 \ldots m$ **do**
$y'_i = \text{argmax}_{y \in GEN(x)} \Phi(x_i, y) \cdot \mathbf{w}$
if $(y'_i = y_i)$ **then**
$c_k = c_k + 1$
else
$\mathbf{w} = \mathbf{w} + \Phi(x_i, y_i) - \Phi(x_i, y'_i)$
$c_{k+1} = 1$
$k = k + 1$
end if
end for
end for
Output: The updated weight parameter vector $\mathbf{w}$.

Predicting Phase
Input:
The list of weight vectors $\langle (\mathbf{w}_1, c_1), \ldots, \mathbf{w}_k, c_k) \rangle$
An unsegmented sentence x.
Calculate:

$$y* = \arg \max_{y \in GEN(x)} (\sum_{i=1}^{k} c_i \Phi(x, y) \cdot \mathbf{w}_i)$$

Output: The voted top ranked candidate $y*$

Figure A.4 – The voted perceptron algorithm.

ever, the incremental weight updating suffers from over-fitting, which tends to classify the training data better, at the cost of classifying the unseen data worse. Also, the algorithm is not capable of dealing with training data that is linearly inseparable.

Voted perceptron. Freund and Schapire (1999) proposed a variant of the perceptron learning approach, called the voted perceptron algorithm. Instead of storing and updating parameter values inside one weight vector, its learning process keeps track of all intermediate weight vectors, and these intermediate vectors are used in the classification phase to vote for the answer. The intuition is that good prediction vectors tend to survive for a long time and thus have larger weight in the vote. Figure A.4 shows the voted perceptron training and prediction phases from (Freund and Schapire, 1999), with slightly modified representation.

The voted perceptron keeps a count c_i to record the number of times a particular weight parameter vector $(\mathbf{w}_i, c_i)$ surviving in the training. For a training example, if its selected top candidate is different from the truth, a new count c_{i+1}, being initialized to 1, is used, and an updated weight vector $(\mathbf{w}_{i+1}, c_{i+1})$ is produced; meanwhile, the original c_i and weight vector $(\mathbf{w}_i, c_i)$ are stored.

Compared with the original perceptron, the voted perceptron is more stable, due to maintaining the list of intermediate weight vector for voting. Nevertheless, to store those weight vectors is space inefficient. Also, the weight calculation, using all intermediate weight param-

Training Phase
Inputs: Training Data $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$; number of iterations T
Initialization: $\mathbf{w} = 0, \gamma = 0, \rho = 0$;
Algorithm:
  for $t = 1 \ldots T$ **do**
    for $i = 1 \ldots m$ **do**
      $y'_i = \text{argmax}_{y \in GEN(x)} \Phi(x_i, y) \cdot \mathbf{w}$
      if $(y'_i \neq y_i)$ **then**
        $\mathbf{w} = \mathbf{w} + \Phi(x_i, y_i) - \Phi(x_i, y'_i)$
      end if
      $\rho = \rho + \mathbf{w}$
    end for
  end for
Output: The averaged weight parameter vector $\gamma = \rho/(mT)$.

Figure A.5 – The averaged perceptron learning algorithm.

eter vectors during the prediction phase, is time consuming.

Averaged perceptron. The averaged perceptron algorithm (Freund and Schapire, 1999) is an approximation to the voted perceptron which, on the other hand, maintains the stability of the voted perceptron algorithm, but significantly reduces space and time complexities. In an averaged version, rather than using $\mathbf{w}$, the averaged weight parameter vector γ over the m training examples is used for future predictions on unseen data, illustrated in Equation A.8.

$$\gamma = \frac{1}{mT} \sum_{i=1 \ldots m, t=1 \ldots T} \mathbf{w}^{i,t} \tag{A.8}$$

where

- $\mathbf{w}$: a weight parameter vector,
- m: a set of training examples,
- T: number of iterations,
- γ: the averaged weight parameter vector.

In calculating γ, an accumulating parameter vector ρ is maintained and updated using $\mathbf{w}$ for each training example. After the last iteration, $\rho/(mT)$ produces the final parameter vector γ. The entire algorithm is shown in Figure A.5.

As mentioned above, the perceptron, especially averaged one, is one of the most powerful model in parsing and in resolving different problems in natural language processing. Collins and Roark (2004) reported that a incremental parsing with the use of averaged perceptron could reach a comparable $Fscore$ (86.6%) comparing to the generative model (86.7%). Zhu et al. (2013) shew that perceptron-based parser could achieve $Fscore$ of 90.4%, which outperformed the currrent state-of-the-art generative parser (Petrov and Klein, 2007) without using any latent variables. Carreras et al. (2008) proposed a way of using Tree Adjoining Grammar (TAG) with the use of perceptron algorithm, which could produce a parsing accuracy of 91.1%, certainly one of the state-of-the-art accuracy in parsing technique.

A.2.5 Advanced parsing methods

Beside the above learning models, there are a number of advanced methods that utilized the external information to boost the performance of parsing systems to higher levels. Socher et al. (2013) used the deep learning technique, which was based on the recurrent neural network, and reach the $Fscore$ of 90.5%. Charniak and Johnson (2005) proposed a general framework called Re-ranking parser. This framework first used a baseline generative parser (such as one in Collins (1999) or in Petrov and Klein (2007)) to produce top k-best candidate parse trees, and then used a discriminative model with a set of strong and rich features to re-rank them and pick out the best one. This work used maximum entropy model as a discriminative re-ranker for the baseline system, which could achieve a high $Fscore$ of 91.5% on test set of English Treebank. Huang (2008) improved the strategy for the re-ranking parsers that could encode more candidate parse trees in the first phase and utilize the averaged perceptron model to perform the re-ranking phase, reaching up to $Fscore$ of 91.8% on English test set.

However that is not whole story, McClosky et al. (2006) even extended the idea of re-ranking parser by injecting more unsupervised features from large external text corpus, making the parser become a self-trained system that could achieve a $Fscore$ of 92.4% on the test set. Currently, the self-trained parser has been considered as the state-of-the-art parsers in terms of $Fscore$ on the English test set.

A.3 Vietnamese classifiers

Classifiers are independent words considered as nouns, which "occupy a special position in the noun phrase, but do not seem to contribute to the meaning of the noun phrase in any definite way" (Kroeger, 2005). The classifier may possibly categorize referents (normally nouns) based on their attribute such as shape, function, or animacy. Unlike European languages, in general, Vietnamese common nouns are required to be accompanied by a classifier, and vice versa since the meaning of a Vietnamese classifier cannot be specified in isolation. Vietnamese is one of several Asian languages with a complex numeral classifier system. In English, most nouns need to be chosen between a singular and a plural (e.g. table vs. tables) whereas Vietnamese nouns "do not in themselves contain any notion of number or amount. In this respect they are all somewhat like English mass nouns such as milk, water, flour, etc." (Thompson, 1987, p. 193). Vietnamese classifiers can be used in "anaphoric construction where classifiers are considered as a pronoun to replace the omitted head noun" (means "one"), such as "cái lớn" (a big one). Two most commonly used classifiers in Vietnamese language rare "con" (for animate, non-human objects) and "cái" (for inanimate objects) (Dao, 2011). Major Vietnamese classifiers are presented as below.

- Animate objects
  - Animals: "con"; e.g. "con chim" *(bird)*, "con bọ hung" *(beetle)*, "con cá" *(fish)*, "con bò" *(cow)*. Some exceptional cases for inanimate objects or others, e.g. "con dao" *(knife)*, "con thuyền" *(boat)*, "con sông" *(river)*, "con nước" *(tide)*.
  - Plants: "cây" *(tree/plant)*, "hoa" / "bông" / "đoá" *(flower)*, "quả" *(fruit)*, "củ" *(root)*, etc.; e.g. "cây bưởi" *(grapefruit tree)*, "cây cà chua" *(tomato plant)*, "hoa hồng" *(rose)*, "hoa sen" *(lotus)*. Some exceptional cases for miscellaneous things, e.g. "cây bút chì" *(pencil)*, "cây nến" *(candle)*.
- Inanimate objects

  - Items: “cái”, “chiếc”, “bức”, “tờ”,. . . ; e.g. “cái bàn” *(table)*, “cái nhà” *(house)*, “cái áo” *(shirt)*, “cái ngõ” *(alley)*, “chiếc dép” *(slipper)*, “chiếc nhẫn” *(ring)*, “bức ảnh” *(picture)*.
  - Three dimensional things: “tảng” *(block)*, “khối” *(plinth)*, “cục” *(clod)*, “toà” *(uplift)*, “ngôi” *(structure)*,. . . ; e.g. “tảng băng” *(iceberg)*, “khối đá” *(plinth of rock)*, “toà nhà”.*(building)*.
  - Flat surfaces/others: “mặt” *(face)*, “mảnh” *(piece)*, “thửa” *(paddy)*, “tờ” *(sheet)*,. . . ; e.g. “mặt bàn” *(table top)*, “thửa ruộng” *(rice paddy)*, “tờ giấy” *(paper sheet)*, “mảnh đất” *(parcel of land)*.

- Human beings
  - With respect: “vị”, “đấng”, “bậc”,. . . ; e.g. “vị anh hùng” *(hero)*, “đấng cứu thế” *(savior)*, “bậc vĩ nhân” *(adorably great person)*.
  - For Ordinary: “kẻ” *(a person without respect)*, “mụ” *(old woman)*, “đứa” *(child)*,. . . ; e.g. “kẻ cắp” *(thief)*, “mụ chủ chứa” *(bawd)*, “đứa con trai” *(son)*.

- Groups
  - Collectives: “bầy”, “đàn”, “đám”, “loạt”, “bó”, “lũ”, “đống”,. . . ., e.g. “đám sinh viên” *(a crowd of students)*.
  - Recurrence: “đợt”, “cơn”,. . . , e.g. “ba đợt sóng” *(three billows of waves)*.

- Miscellaneous things
  - Emotions: “nỗi” *(feeling)*, “niềm” *(sentiment)*,. . . ; e.g. “niềm vui” *(joy)*, “nỗi buồn” *(sadness)*, “niềm hy vọng” *(hopefulness)*, “nỗi đau” *(affliction)*.
  - Events: “cuộc”, “trận”, “nạn”, “vụ”,. . . ; e.g. “trận bóng đá” *(football match)*.
  - Others: “bài”, “bản”,“cuốn”,. . . , e.g. “ba đợt sóng” *(three billows of waves)*.

Appendix B

Corpus design and prosodic phrasing modeling

Contents

B.1 Semi-automatic correction of breath noise labeling

The synthetic voice trained with the first result of the annotated corpus sounded discontinuous at some transitions between syllables. Some observations were done on the labeled speech of the training corpus for discovering the problems. Although EHMM could deal with pause insertion, but it often failed to predict the pause appearance or pause duration in the speech corpus since the speaker mainly produced breath noises instead of silence pauses. This issue seemed to appear more frequently if the previous or next phones had either similar phonetic features as a breath noise (e.g. [h]), or unvoiced signals (e.g. /p t k f v z/).

Figure B.1 and Figure B.2 illustrates several examples of such problems in the labeling results. A number of breath noises were not segmented/assigned with a separate label (i.e. "pau"). They were confused with the adjacent segments. In some cases, a part of the breath noise could also be wrongly labeled, as illustrated in Figure B.2 (a). In this case, there was a separate "pau" label, yet with a much smaller duration.

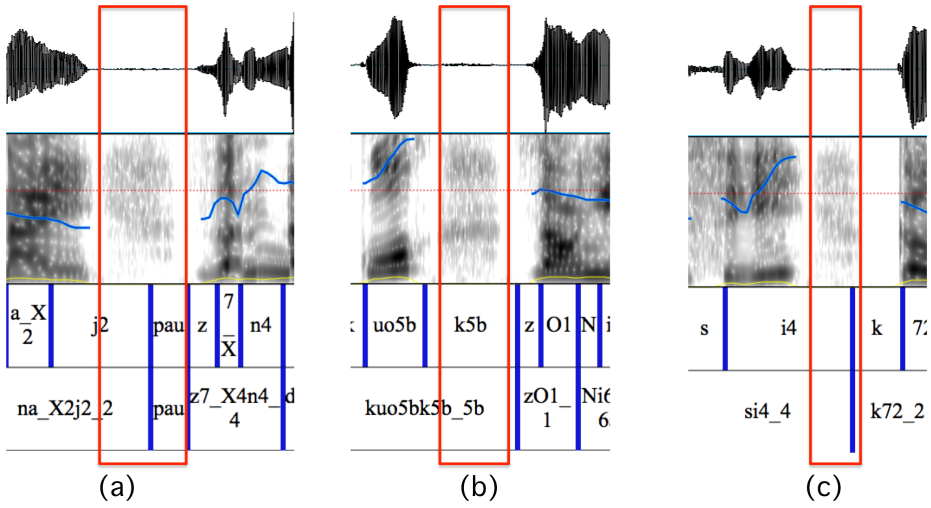


Figure B.1 – Breath noises were wrong labeled as a part of the previous segments [j k i] in the carrying text: (a) do nghị sĩ Chang Young Dal, và đoàn Nga [], (b) nghị sĩ klyus viktor - alexandrovich, (c)

Automatic correction process. After doing some observations and measurement on the speech signals, we discovered a process to correct these wrong labeling. The general solution is summarized as below.

- Setting some duration limits for phone types: consonant or vowel, long or short lexical tone (by signal observations)
- Calculating the mean value and the standard deviation for each phone using its samples whose durations did not exceed the respective duration limits.
- Assuming that the pseudo-duration of a phone was the total of its mean value and its standard deviation
- For each segment in the speech corpus, if its duration was more than twice its pseudo-duration and the segment was the first or the last of the bearing syllable, it was considered as a wrong labeling. There were two cases for such problem:

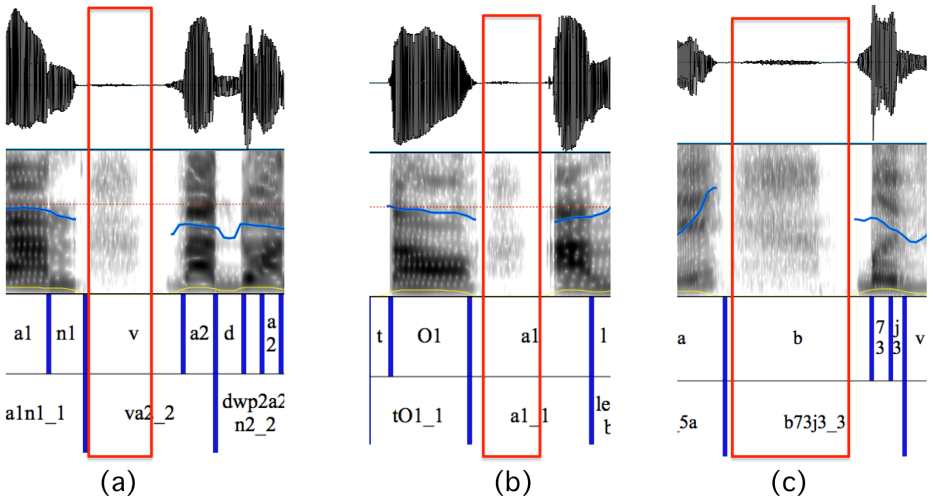


Figure B.2 – Breath noises were wrong labeled as a part of the next segments [v a b].

- *Breath noises were wrongly labeled as a part of the previous segment A (Figure B.1)*: Breath noise (pause) + Segment
  - Reduced the duration of A = The pseudo − duration of A
  - Added a new or modified the existing pause (if exists) **succeeding** A with a duration = Old duration of A − New duration of A
- *Breath noises were wrongly labeled as a part of the next segment B (Figure B.2)*: Segment + Breath noise (pause)
  - Added a new or modified the existing pause (if exists) **preceding** B with a duration = Old duration of B − Pseudo-duration of B
  - Reduced the duration of B = The pseudo − Duration of B

Manual correction. Several manual refinements for some special phones (e.g. long vowels, semi-vowels) were investigated to get a better result. We corrected many cases where pauses were too small, or words had only a single vowel syllables, etc.

B.2 VNSP-ThuTrang

A total of 630 sentences in the existing text corpus VNSP, as presented, were also recorded by the speaker Thu-Trang for comparison. This subsection gives some comparisons between the two speech corpora with a similar text content but recorded by different speakers and recording condition. The old corpus was recorded in Vietnam by a female broadcaster from Hanoi at 16 kHz and 16 bps (hence called VNSP-Broadcaster); meanwhile the new one was recorded in France by a female non-professional speaker Thu-Trang from Hanoi at 48 kHz, 24 bps and eventually converted to 48 kHz, 16 bps (hence called VNSP-ThuTrang).

Detailed information of these two audio corpora can be found in Table B.1. There were 630 utterances in both corpora, actual numbers of syllables/segments nevertheless were slightly different (8,479 syllables in VNSP-ThuTrang and 8,465 syllables in VNSP-Broadcaster), because some non-standard words (NSWs, e.g. date, time, numbers) were read differently by the two speakers. The broadcaster irregularly spoke lacking of the full expansion of NSWs, e.g. “hai nghìn linh một” *(two thousands and one)*, instead of “hai nghìn không trăm linh

một" *(two thousands zero hundred and one)* for the year "2001"; "ngày chín tháng một" instead of "ngày **mùng** chín tháng một" for the date "09/1" (with dates having days less than 10, "mùng" [muŋ-2] always precedes the days in Vietnamese).

Table B.1 – VNSP corpus with Broadcaster and ThuTrang voices

Analysis factor	VNSP-ThuTrang	VNSP-Broadcaster
Number of utterances	630	630
Number of segments	22,919	22,877
Number of syllables	8,479	8,465
Number of perceived pauses	943	790
Total duration (minutes)	39.73	31.60
Speech rate (segments/s)	9.61	12.06
Speech rate (syllables/s)	3.56	4.46
Total duration without pauses (minutes)	35.66	28.71
Speech rate without pauses (segments/s)	10.71	13.28
Speech rate without pauses (syllables/s)	3.96	4.91

The speech rate of the VNSP-Broadcaster was about 2.5 segments/s or 25% faster, hence over 8 minutes or 20% shorter than that of VNSP-ThuTrang. The number of perceived pauses made by Thu-Trang was greater than by the professional speaker about 16% (more 153 pauses).

B.3 Syntactic rules

B.3.1 Formal symbols representing syntactic rules

For representing syntactic rules, formal symbols were proposed as follows. These formal symbols were used to express syntactic rules for further automatic processing in boundary prediction and fine-tuning.

- $A|B$: A or B
- $A + B$: B follows A
- . : Any character
- ; : List of elements
- $-$: There is a prosodic boundary
- A'*abc*' : The element A having text 'abc'
- $A = x$: The element A having x syllables
- $A >= x$: The element A having at least x syllables
- $A <= x$: The element A having at most x syllables
- $A > x$: The element A having more than x syllables
- $A < x$: The element A having less than x syllables

- $A(child : B|C)$: The element A having a child B or C
- $A(: B)$: Constituent A is dependent B or vice versa
- $[abc]$: One character among the list in square brackets
- $a\{x, y\}$: The character a appears from x to y times, the default value for y is any (i.e. $.\{1,\}$ for any number of any character)

Some formal symbols were adopted from regular expressions, e.g. "|" for "*or*", "." for "any character", "[*abc*]" for "any character in the list in the square bracket", "{x,y}" for the number of appearance. There were several new symbols, specified for boundary ("−"), ordered sequence of syntactic elements ("+"), list of syntactic elements (";"), lower and upper limits of number of syllables ($>= x$, $> x$, $<= x$, $< x$, $= x$). Symbols for elements themselves (":"), their parents(": *parent*") and their child(": *child*") were also given. Some special syntactic elements could be specified by their content ('*abc*').

B.3.2 Proposal of syntactic rules

After having observed relations between syntax and pause appearance in the corpus, we proposed some hypotheses on syntactic rules with corresponding break levels. Two types of rules were discovered: (i) Constituent syntactic rules between two constituents in phrase structure grammar and (ii) Functional rules between two dependents in relational structure. Proposed rules with syntactic constituents and with syntactic dependents are presented respectively in Table B.2 and Table B.3.

The highest break level in middle of sentences ("4") were set if either the left constituent is or contains a clause (S, SB), i.e. the rules HC1 and HC2, or both left and right dependent elements were predicates (PRD), i.e. the rule HD1, or head elements (H), i.e. the rule HD2. Other decisions were made on the basis of syntactic element names (e.g. adjuncts ADT) or/and number of syllables in the left or right elements. Smaller break levels ("2" and "3") may appear after some special POS or syntactic phrases, e.g. prepositional phrases PP, conjunction C. Syntactic rules were refined using number of syllables, parents or children of syntactic elements. For instance, we found that there was a boundary between a phrase having at least 7 syllables, and a phrase having at least 4 syllables (HC3). These number limits were optimized through several iterations from proposal to evaluation. We describe hereafter more detail with some examples for each syntactic rule shown in the two tables.

Constituent syntactic rules. Constituent syntactic rules in Table B.2 are described in detail as follows.

HC1: $SB; .\{1,\}(child : S|SB)-$ means that there is a boundary between a subordinate clause (SB) or any constituents having a clause child (child:S|SB) AND any constituent. E.g. "[Người đàn ông [mà bà gặp hôm qua ở nhà tôi]$_{SB}$]$_{NP}$ − là một người rất tốt bụng" *([The man [you met yesterday at my home]$_{SB}$]$_{NP}$ − is a really kind person).*

HC2: $S >= 6-$ means that there is a boundary between a main clause having at least 6 syllables (S>=6) AND any constituent. E.g. "[Mùa thu lá vàng rơi đầy trên từng góc phố]$_S$) − còn mùa xuân thì hoa nở muôn nơi" *([In autumn yellow leaves fall down all over streets]$_S$ − and in spring flowers bloom everywhere).*

HC3: $.\{1,\}P >= 7-.\{1,\}P >= 4$ means that there is a boundary between a phrase having at least 7 syllables AND a phrase that having 4 syllables. E.g. "Nhưng [kết cấu của

Table B.2 – Constituent syntactic rules

Break level	Rule code	Formal representation	Left element	Right element
4	HC1	$SB; .\{1,\}(child : S\|SB)-$	a subordinate clause or any constituents having a clause child	any constituent
	HC2	$S >= 6-$	a main clause having at least 6 syllables	any constituent
3	HC3	$.\{1,\}P >= 7-.\{1,\}P >= 4$	a phrase having at least 7 syllables	a phrase having at least 4 syllables
	HC4	$.\{1,\}P-C + .\{1,\} >= 5$	a phrase	a conjunction whose next constituent having at least 5 syllables
	HC5	$PP >= 3-C; [ANV]P$	a prepositional phrase having at least 3 syllables	a conjunction or an adjective/noun/verb phrase
	HC6	$3 <= S <= 5$-	a main clause having 3 or 5 syllables	any constituent
	HC7	$C(parent : SB)$'rằng'$-$	the conjunction 'rằng' having a parent that is a subordinate clause	any constituent

loại máy bay ba lớp cánh này]$_{NP}$ − rất phức tạp và khó chế tạo" *(However [the structure of this 3-wing-tier airplane kind]$_{NP}$ − is very complicated and difficult in production).*

HC4: $.\{1,\}P-C+.\{1,\} >= 5$ means that there is a boundary between a phrase (.{1,}P) AND a conjunction (C) whose next constituent (+) having at least 5 syllables (.{1,}>=5). E.g. "Cậu [luôn ý thức bảo vệ mẹ mình]$_{VP}$ − [và]$_C$ [thường xuyên giúp đỡ bạn bè trong lúc khó khăn]$_{VP}$" *(He [always stays up conscious to protect his mother]$_{VP}$ − [and]$_C$ [usually helps his friends in difficulties]$_{VP}$).*

HC5: $PP >= 3-C; [ANV]P$ means that there is a boundary between a prepositional phrase having at least 3 syllables (PP>=3) AND a conjunction (C) or an adjective/noun/verb phrase ([ANV]P). E.g. "Đó là kết quả của những buồn vui [trong tình yêu của riêng mình]$_{PP}$ − [và]$_C$ cả những tâm sự của khán giả dành cho tôi" *(That is the results of joyfulness and sadness [in their own love]$_{PP}$ − [and]$_C$ also their confidings given to me).*

HC6: $3 <= S <= 5-$ means that there is a boundary between a subordinate clause or any constituents having a clause child AND any constituent. E.g. "[Công trình đủ thợ]$_S$ − thì các chị đỡ vất vả hơn" *(There is enough workers for the construction - then these women are less hard-working)*; [Năm tháng qua đi]$_S$ − [họ lớn lên]$_S$ - mỗi người một hoàn cảnh] *([The time goes by]$_S$ − [they grew up]$_S$ − each person has their own situation).*

HC7: $C(parent : SB)$**'rằng'**$-$ means that there is a boundary between the conjunction 'rằng' (C'rằng') having a parent that is a subordinate clause (parent:SB) AND any constituent. E.g. "Chúng tôi cho [[rằng]$_C$ − những anh chị nên về nước để xây dựng tổ quốc]$_{SB}$" *(We suppose [[that]$_C$ − these people should come back to develop their country]$_{SB}$).*

Functional rules. Functional rules in Table B.3 are described in detail as follows.

Table B.3 – Functional rules

Break level	Rule code	Formal representation	Left element	Right element
4	HD1	PRD−PRD	a predicate	a predicate
	HD2	ADT>=3−.{1,}(:.{1,}P)	an adjunct having at least 3 syllables	any dependency element that is a phrase
	HD3	H>=4−H	a head element having at least 4 syllables	a head element
2	HD4	2<=H<=3−H	a head element having 2 or 3 syllables	a head element
	HD5	2<=ADT<=3−SUB>=2	an adjunct having 2 or 3 syllables	a subject having at least 2 syllables

HD1: $PRD{-}PRD$ means that there is a boundary between two predicates. E.g. "Lão [có nhà ở ngoại ô]$_{PRD}$ − [có ô tô hạng sang và vài người giúp việc]$_{PRD}$" *(He [had a house in a suburban area]$_{PRD}$ − [had a luxury car and several housekeepers]$_{PRD}$).*

HD2: $ADT >= 3{-}.\{1,\}(: .\{1,\}P)$ means that there is a boundary between an adjunct (ADT) having at least 3 syllables (>=3) AND any dependency element that is a syntactic phrase (:.{1,}P). E.g. "[Với mức lương 1000 USD một tháng]$_{PP-ADT}$ − Văn có thể sống khá thoải mái cùng gia đình ở thành phố lớn" *([With the salary of USD 1000 per month]$_{ADT}$ − [Van]$_{NP}$ can live comfortably with his family in a big city).*

HD3: $H >= 4{-}H$ means that there is a boundary between a head element having at least 4 syllables AND a head element. E.g. "Quán này có rất nhiều món ngon như [Cơm Tám Mễ Trì]$_{NP-H}$ − [cơm rang các món]$_{NP-H}$ − [các món nhậu về cá]$_{NH-H}$" *(This restaurants have many delicious dishes, such as [the Tam Me Tri rice]$_{NP-H}$ − [varied dishes of fried rice]$_{NP-H}$ − [fish dishes for carouse]$_{NP-H}$).* "Chúng ta thiếu những cán bộ [năng lực chuyên môn cao]$_{NP-H}$ − [có uy tín quốc tế]$_{VP-H}$ để tiến hành những nghiên cứu lớn" *(We are lacking of researchers with [high quality specialty]$_{NP-H}$ − [good international impact]$_{NP-H}$ to perform critical researches).*

HD4: $2 <= H <= 3{-}H$ means that there is a boundary between a head element having 2 or 3 syllables (2<=H<=3) AND a head element (H). E.g. "Những ca khúc của Phương Uyên mang tính tự sự − lắng đọng" *(Phuong Uyen's songs is [self-confiding]$_{A-H}$ − [speechless]$_{A-H}$);* "Mới 13 tuổi mà hắn đã là một tên gián điệp [lợi hại]$_{A-H}$ − [ranh ma]$_{A-H}$ − [quỷ quyệt]$_{A-H}$" *(Only 13 years old but he already was a [skillful]$_{A-H}$ [artful]$_{A-H}$ [crafty]$_{A-H}$ spy).*

HD5: $2 <= ADT <= 3{-}SUB >= 2$ means that there is a boundary between an adjunct having 2 or 3 syllables (2<=ADT<=3) AND a subject having at least 2 syllables (SUB>=2). E.g. "[Đột nhiên]$_{R-ADT}$ − [lão]$_{NP-SUB}$ bán tất để chuyển vào thành phố" *([Suddenly]$_{R-ADT}$ − [he]$_{NP-SUB}$ sold everything in order to move to city).*

B.4 Breath groups and syllable ancestors

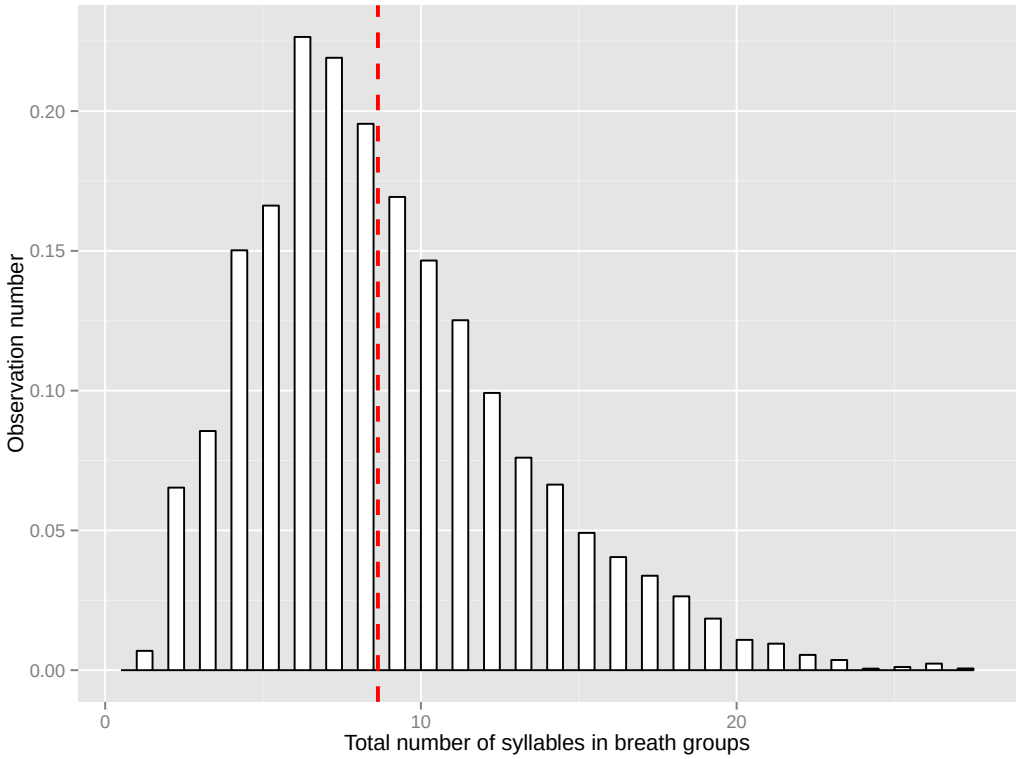


Figure B.3 – Distribution of Breath Group length in VTDO-Analysis.

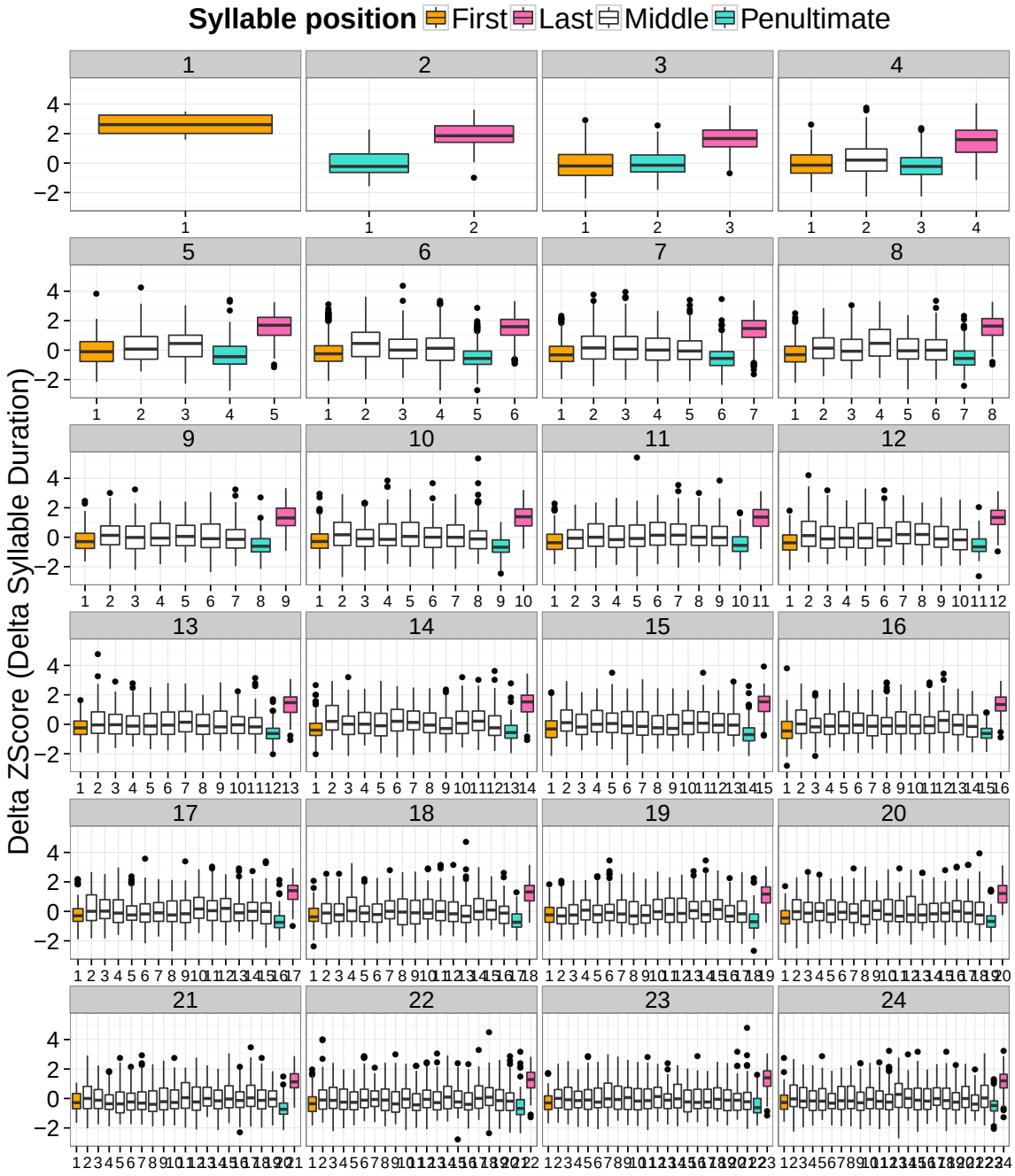


Figure B.4 – Distributions of syllables' durations (ZScore) by positions in highest ancestors in the VTDO-Analysis corpus, factored by syllable numbers of highest ancestors. Ancestors having more than 24 syllables were excluded.

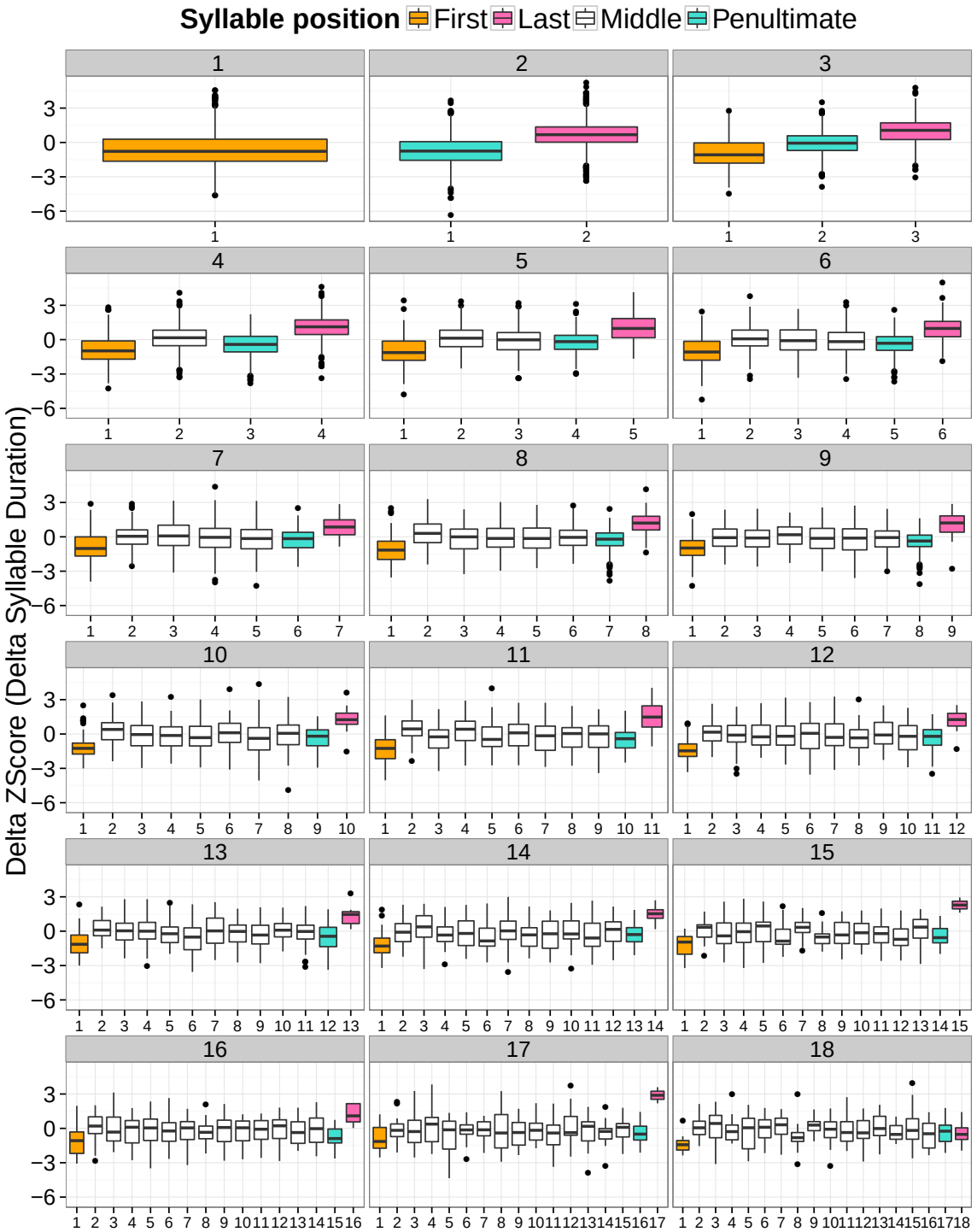


Figure B.5 – Distributions of syllable duration differences (Delta ZScore) by positions in lowest ancestors in the VTDO-Analysis corpus, factored by syllable numbers of these ancestors. Last syllables of higher level ancestors and syllables with subsequent pauses were excluded. Ancestors having more than 24 syllables were excluded.

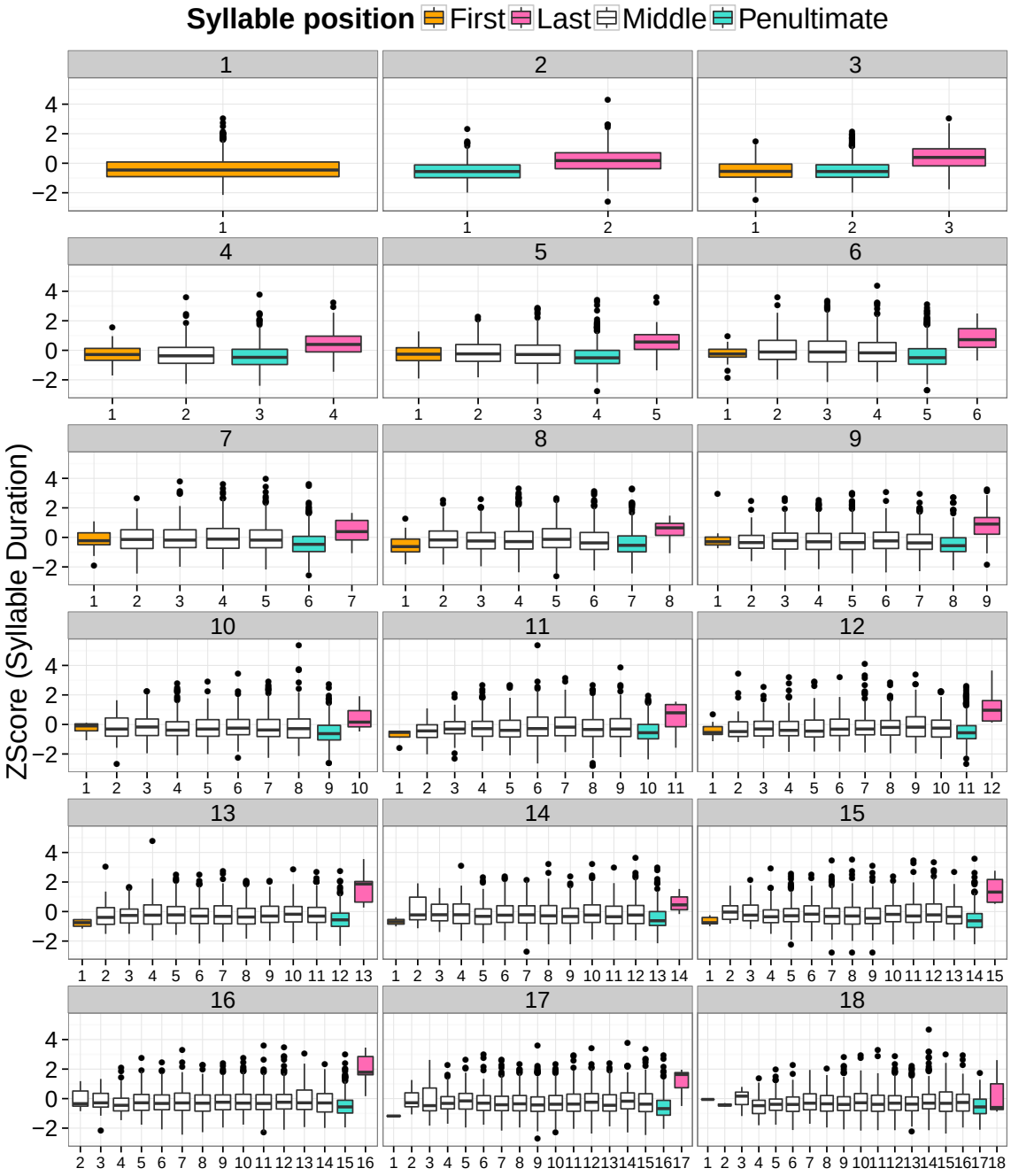


Figure B.6 – Distributions of syllable durations (ZScore) by positions in syntactic blocks with a maximum of 27 syllables, factored by syllable numbers of these blocks. Syllables with subsequent pauses were excluded. Ancestors having more than 18 syllables were excluded.

B.5 Syntactic blocks

Figure B.7 depicts the distributions of syllable duration (ZScore) of final syllables of syntactic blocks, factored by syllable numbers of these blocks. It was clear that after syntactic blocks having at least 2 syllables, there was a final lengthening, one cue of prosodic phrasing. The level of lengthening was higher at the end of syntactic blocks with at least 3 syllables.

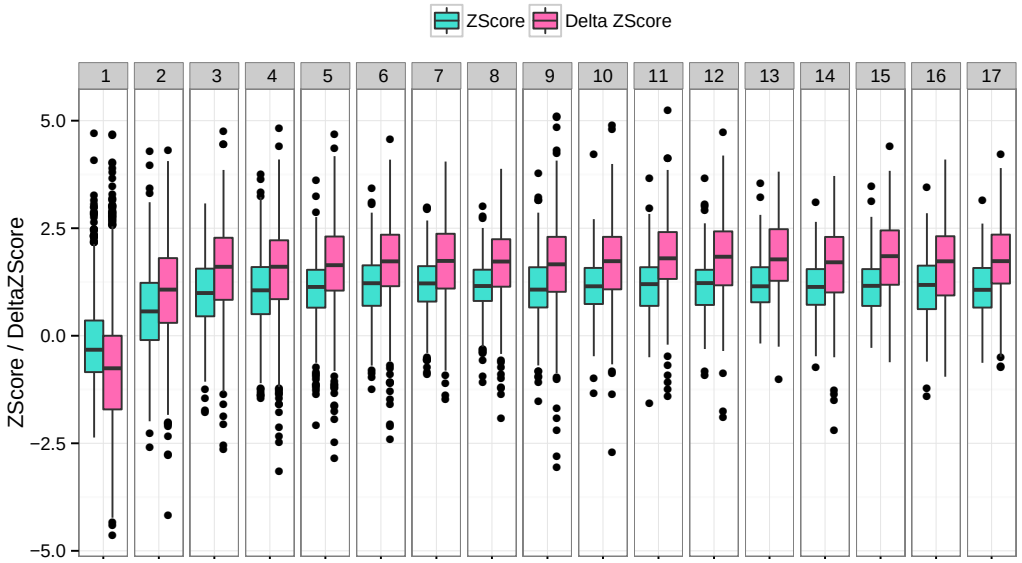


Figure B.7 – Distributions of duration (ZScore normalization) of final syllables of syntactic blocks with a maximum of 17 syllables, factored by syllable numbers of these blocks.

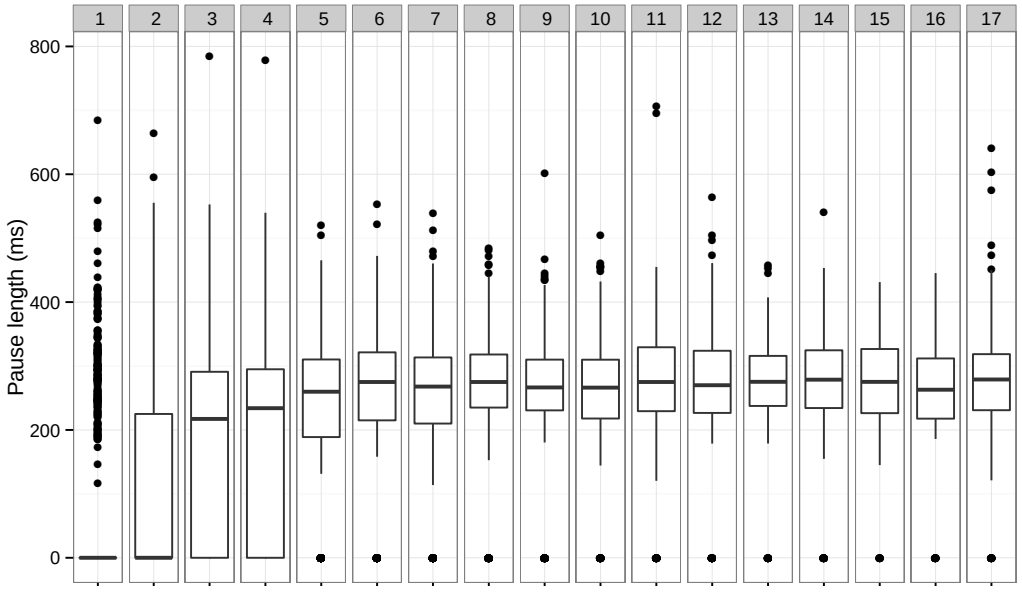


Figure B.8 – Distributions of pause length of final syllable of syntactic blocks with a maximum of 17 syllables, factored by syllable numbers of these blocks.

For another stronger cue of prosodic phrasing, i.e. pause appearance, we examined actual pauses in the middle of utterances after syntactic blocks. Pause presence could be roughly predicted by syntactic block size: pauses mostly appeared at the end of syntactic blocks having at least 5 syllables, as shown in Figure B.8. Based on the median of pause length in the figure, we supposed only one level of pause length. As a result, we proposed here two levels of prosodic phrasing for the Vietnamese TTS, i.e. final lengthening and pause appearance.

B.6 Algorithm of syntactic-block devision

Full algorithm for dividing one syntactic node (parent) to syntactic blocks with a limit size (limitSize = 6 for lengthening) is presented in Listing B.1. To get syntactic blocks for a sentence, we pass the root of that sentence and the limit size of phrases.

Listing B.1 – Identification process of syntactic blocks for final lengthening

```
/**
     * Get list of descendant syntactic blocks whose maximal syllable number is
         limitLength, combining single syllable blocks
     * Each syntactic phrases include list of descendant words List<Node>
     * @RETURN phrases
               list of syntactic blocks List<Phrase>
     * @PARAM ancestor
               the ancestor node of all returned descendant syntactic blocks
     * @PARAM limitLength
               the limit syllable number of descendant syntactic blocks
*/
List<Phrase> getSyntacticBlocksLengthening(Node ancestor, int limitLength){
   INITIALIZE results AS a List<Phrase>
   IF syllable number of ancestor <= limitLength {
      RETURN ancestor;
   } ELSE {
      INITIALIZE oneSyllablePhrase AS Phrase;
      FOR EACH child OF ancestor {
         IF syllable number of child > 1 {
            phrasesOfChild = getSyntacticBlocksLengthening(child, limitLength);

            IF oneSyllablePhrase HAS more than 1 child {
               ADD oneSyllablePhrase TO results;
               ADD ALL phrases IN phrasesOfChild TO results;
               RETURN results;
            } ELSE IF oneSyllablePhrase HAS 1 child {
               //combine to the next phrase
               firstPhrase_Child = GET the first phrase OF phrasesOfChild;
               ADD oneSyllablePhrase TO firstPhrase_Child;
               ADD phrasesOfChild TO results;
            }

            IF ALL next children ARE one-syllable phrases {
               lastPhrase_ancestor = all next children;
               IF lastPhrase_ancestor HAS more than 1 child {
                  ADD ALL phrases IN phrasesOfChild TO results;
                  ADD lastPhrase_ancestor TO results;
                  RETURN results;
               } ELSE IF lastPhrase_ancestor HAS 1 child {
```

```
                //combine to the last element of ancestor
                lastPhrase_Child = GET the last element OF phrasesOfChild;
                ADD lastPhrase_ancestor TO lastPhrase_Child;
                ADD phrasesOfChild TO results;
             }
          ELSE {
             ADD phrasesOfChild TO results;
          }
       ELSE {
          ADD child TO oneSyllablePhrase;

          IF child IS the last one OF ancestor
             ADD oneSyllablePhrase TO results;
             RETURN results;
          }
       }
    }

    return results;
  }
}
```

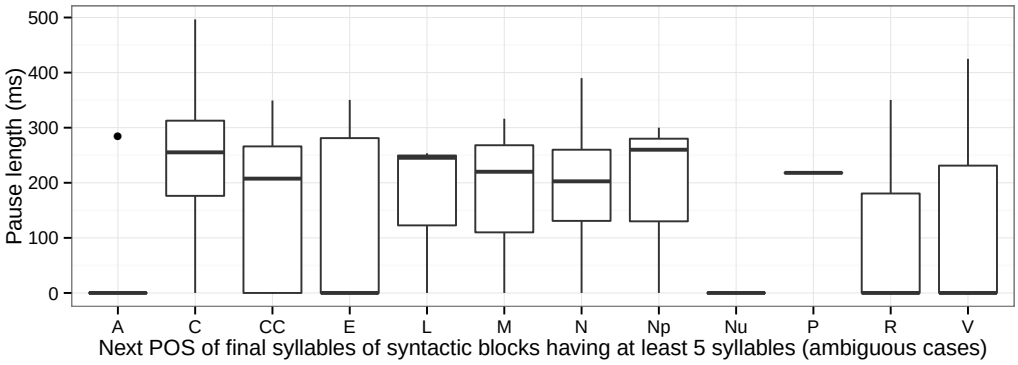
B.7 Syntactic-block+link+POS model

For further improvement, we did find that POSs of the last word (current POS) and that of the next word (next POS) of syntactic blocks could be used to predict pauses with ambiguity. By a preliminary analysis, we supposed that next POSs provided a better clue than current POSs to clarify those cases. Figure B.9 shows distributions of pause length of syntactic blocks having (a) at least 5 syllables (T1) and (b) from 2 to 4 syllables (T2), factored by next POSs. The next POS was a really effective factor when nearly two third of whose distributions clearly separated from zero points (no pause). Few pauses appeared if the next POS was one of those values “A”, “E”, “Nu”, “R” and “V”.

Based on the proposal of the new feature “syntactic link”, we assumed that the syntactic link between the next word and the current word (that is the next syntactic link of the current word) gave a good information for their juncture. If the next word and the current word were loosely linked enough, there might be a pause between them. Figure B.10 illustrates the distributions of pause length after the last syllables of syntactic blocks having at least 2 syllables by this factor. Most pauses appeared after the last syllables if the next syntactic link of the last syllables was “4”, the most loosest value. For less looser next syntactic links, i.e. “2” and “3”, there were some ambiguous that need to be clarified.

Some ambiguous cases, i.e. next POSs of “CC” in T1, or “L,M,T” in T2, could be elucidated by combining next POSs and current POSs. Distributions of pause lengths for those cases were discovered in Figure B.11. Ambiguous cases or cases with few pauses were removed.

(a) T1



(b) T2

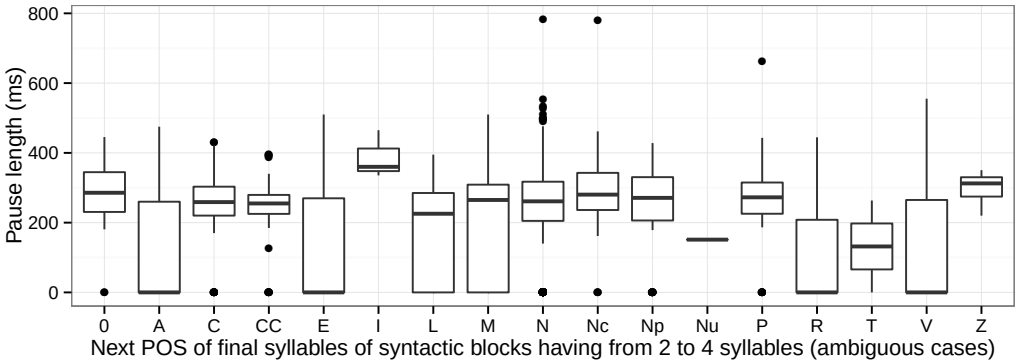


Figure B.9 – Distributions of pause length after the last syllables of syntactic blocks having (a) at least 5 syllables (ambiguous cases with next and current syntactic links of 2-2,2-3,3-2,3-4); (b) from 2 to 4 syllables (ambiguous cases with next and current syntactic links of 2-1,2-2,2-h2,3-1,3-l1,4-2). The x-axis shows next POSs of these last syllables.

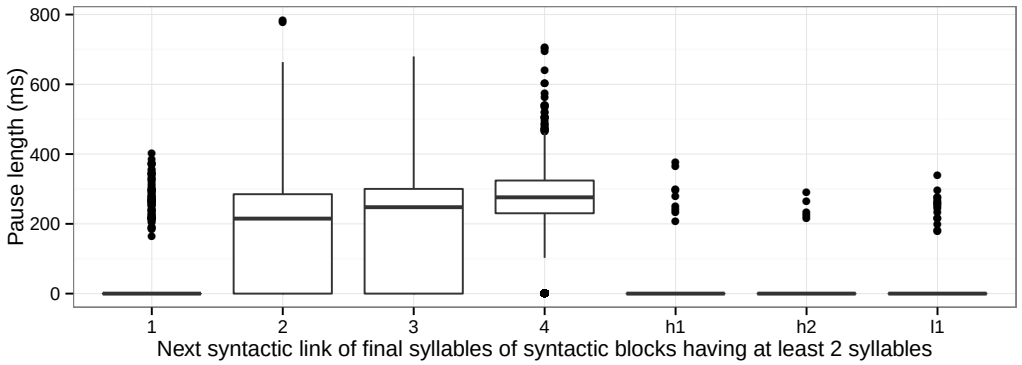
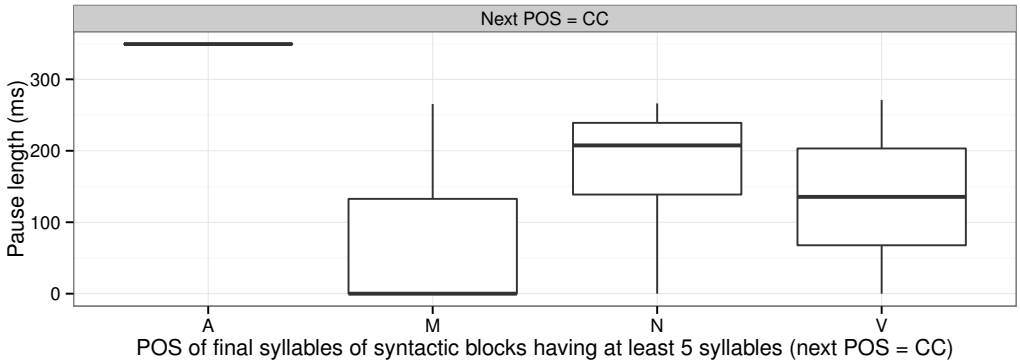


Figure B.10 – Distributions of pause length after the last syllables of syntactic blocks having at least 2 syllables. The x-axis shows the next syntactic link of these last syllables.

(a)



(b)

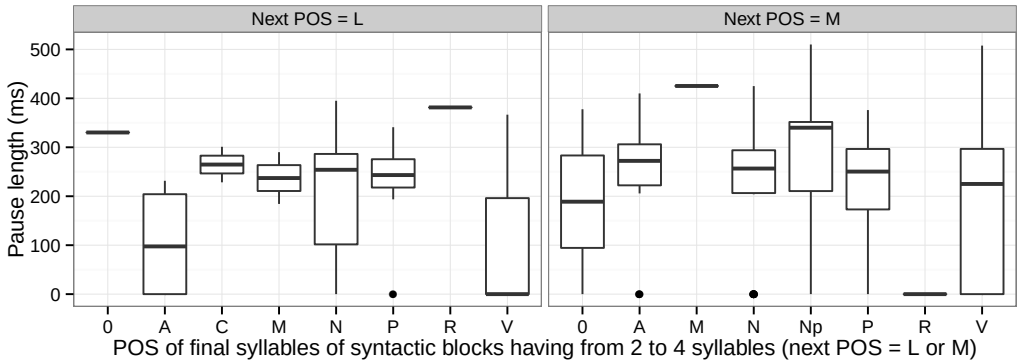


Figure B.11 – Distributions of pause length after the last syllables of syntactic blocks having (a) at least 5 syllables (ambiguous cases with next POSs of "CC"); (b) from 2 to 4 syllables (ambiguous cases with next POSs of "L" or "M"). The x-axis shows the POS of these last syllables, factored by next POSs.

Appendix C

VTED design, construction and perceptual evaluations

Contents

C.1 The ToBI transcription model

In ToBI model, the utterances are described by labels structured in tiers: the orthographic tier, the miscellaneous tier (for comments of all kinds), the break tier (which describes the utterance's phrasing) and, of course most importantly, the tone tier. This subsection consists of two short descriptions of the individual elements of the ToBI tone inventory, which closely follows the example of the ToBI Annotation Conventions (Beckman and Hirschberg, 1994): (i) Prosodic phrasing (ii) Phrasal tones and pitch accents.

Break indices represent a rating for the degree of juncture perceived between each pair of words and between the final word and the silence at the end of the utterance. They are to be marked after the all words that have been transcribed in the orthographic tier. All junctures - including those after fragments and filled pauses - must be assigned an explicit break index value; there is no default juncture type. Values for the break index are chosen from the following set:

- **0:** for cases of clear phonetic marks of clitic groups.
- **1:** most phrase-medial word boundaries.
- **2:** a strong disjuncture marked by a pause or virtual pause, but with no tonal marks; or a disjuncture that is weaker than expected at what is tonally a clear intermediate or full intonation phrase boundary.
- **3:** intermediate intonation phrase boundary; i.e. marked by a single phrase tone affecting the region from the last pitch accent to the boundary.
- **4:** full intonation phrase boundary; i.e. marked by a final boundary tone after the last phrase tone.

The intonation is transcribed as a series of pitch accents and boundary tones each of which can be either low (L), or high (H). Accents are distinguished by appending a star (*), whereas tones are distinguished by appending either a percentage sign (%) or a minus sign (-), denoting boundary and phrase tones, respectively. By tagging individual syllables with these labels, it became possible to identify perceived prominences and major phrase boundaries by * and %, respectively, while the H and L portions of the labels described the shape of the pitch-track. The pitch-track was further described by the use of the "!" diacritic to indicate down-stepping, and the inclusion of the HiF0 label to mark the location of the peak F0 value in each major phrase.

Phrasal tones will be assigned at every intermediate or intonation phrase: L- or H- (phrase accent); L% or H% (final boundary tone) and %H (high initial boundary tone). Since intonation phrases are composed of one or more intermediate intonation phrases plus a boundary tone, full intonation phrase boundaries will have two final tones, e.g. L-L%, L-H%, H-H% and H-L%. Pitch accent tones will be marked at every accented syllable. Lack of pitch accent assignment for a syllable will be interpreted as meaning that the syllable is NOT accented. The ToBI transcription allows for the five types of pitch accents: H* (peak accent), L* (low accent), L*+H (scooped accent), L+H* (rising peak accent) and H+!H* (a clear step down onto the accented syllable from a high pitch).

C.2 Mary TTS platform

Figure C.1 shows the overall process that supports to add a new language to Mary TTS.

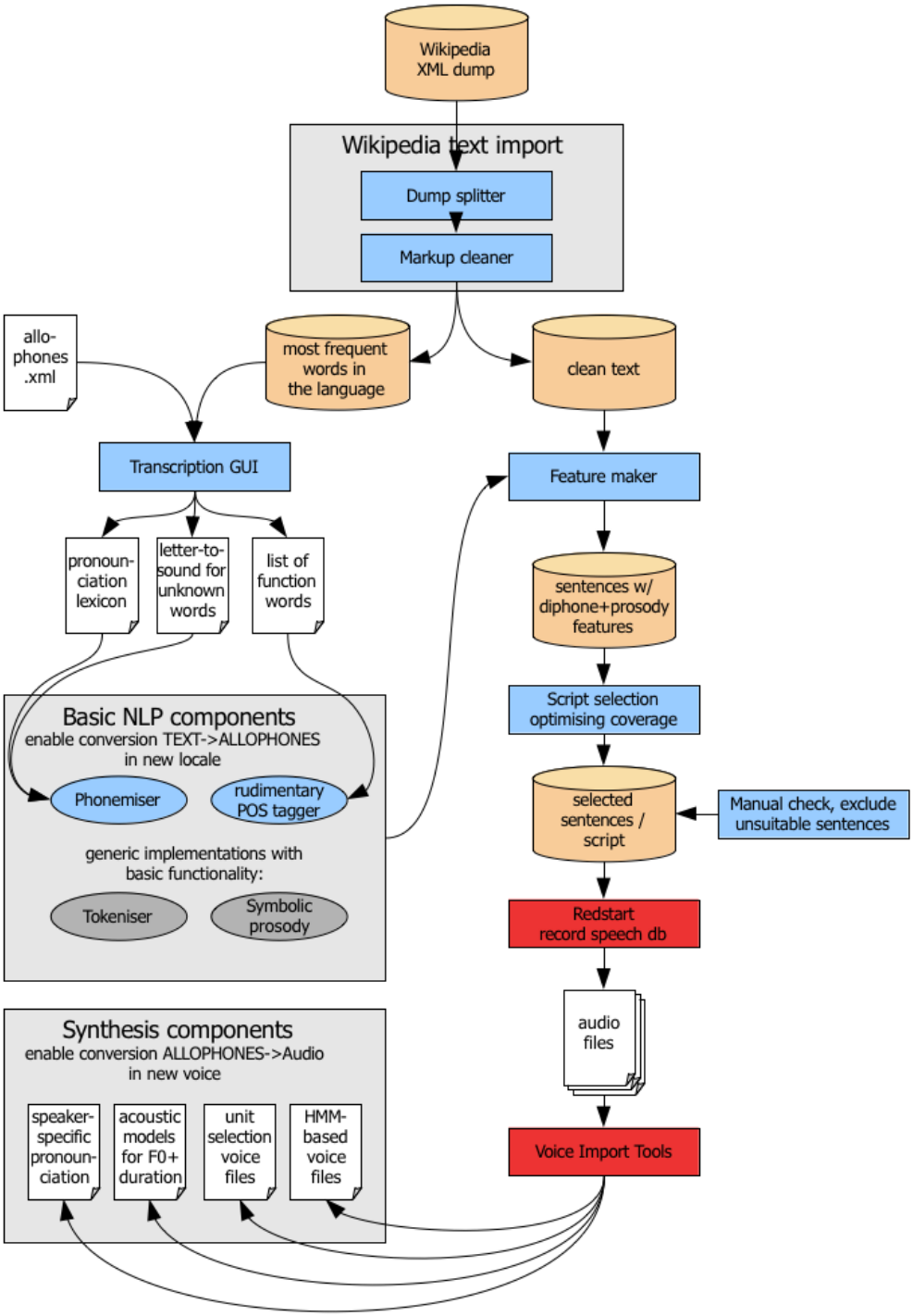


Figure C.1 – Overall process to add a new language to Mary TTS.

C.3 Examples of test GUI screens

This section provides some example GUI screen of our test tool, VEVA, for different perception tests.

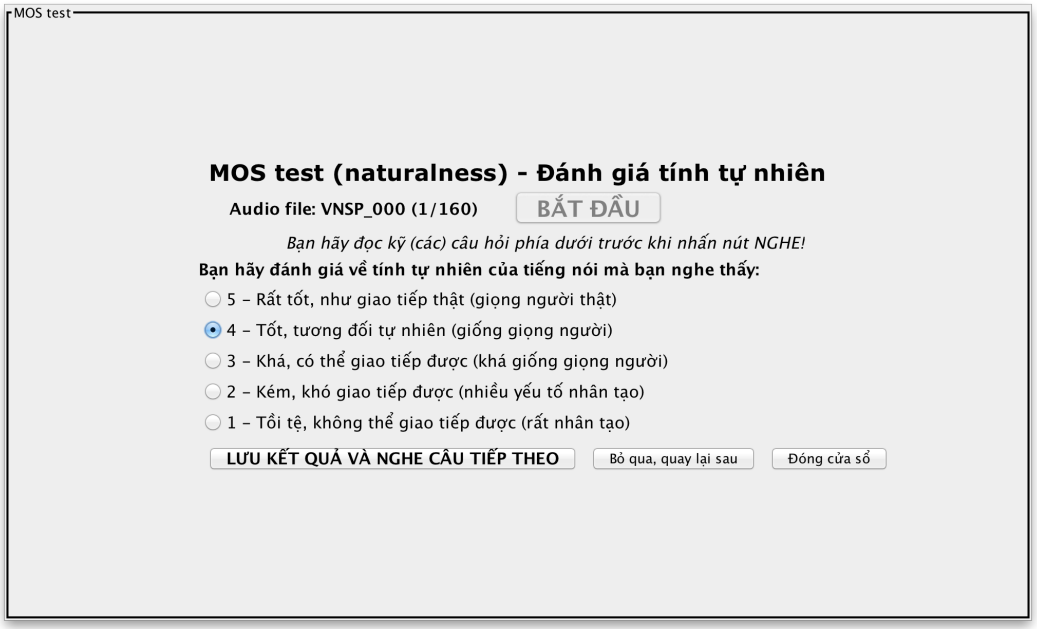


Figure C.2 – GUI of MOS test (naturalness).

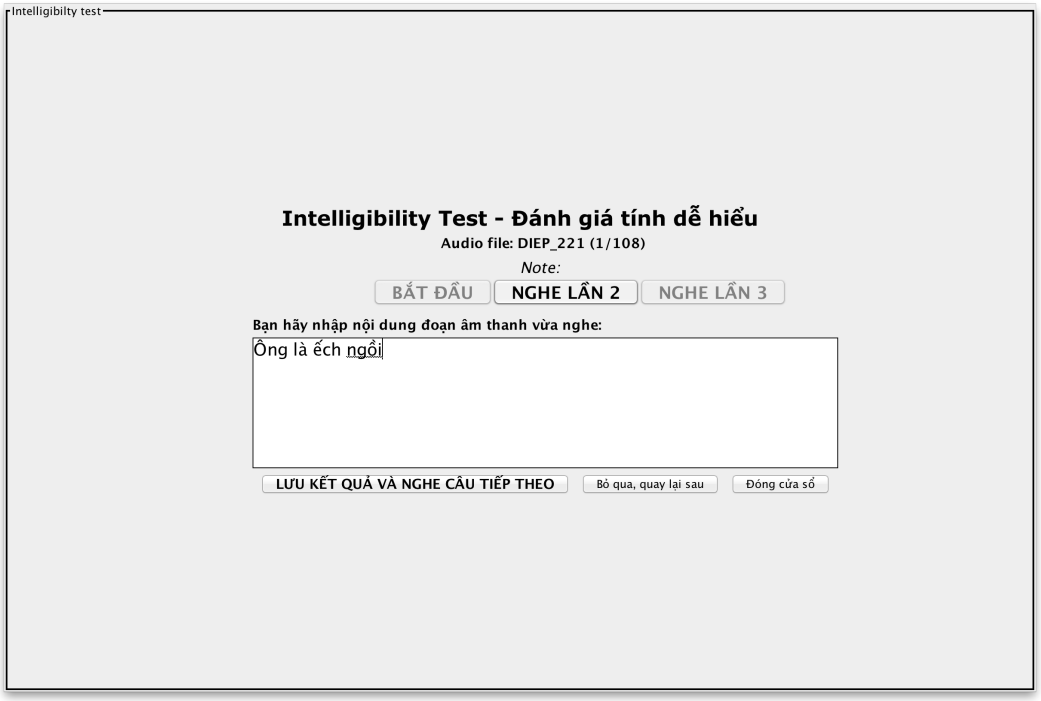


Figure C.3 – GUI of Intelligibility test.

Figure C.4 – GUI of Tone intelligibility test.

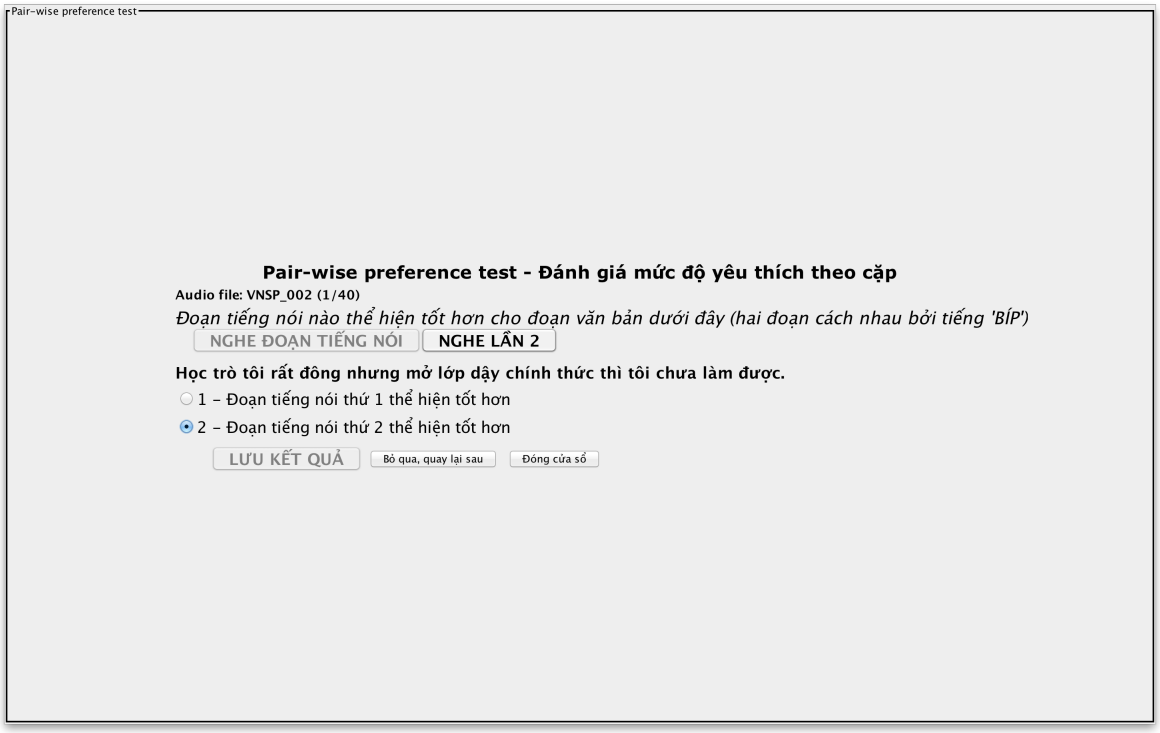


Figure C.5 – GUI of Pair-wise preference test.

C.4 Test corpus examples

This section provides some examples of the test corpus for different perception test. Table C.1 illustrates 14 sample sentences in five groups for tone intelligibility test. Table C.2, C.3, C.4 and C.5 respectively shows 12 sample sentences for MOS, intelligibility, pair-wise preference using syntactic rules or syntactic-blocks+links+POSs. For ease of observation, transcriptions of example sentences in following tables are presented in allophones. The lexical tones are separated by a hyphen "-". Some first sentences in these tables are available with audio files in Nguyen (2015a).

Table C.1 – Test corpus examples of Tone intelligibility test

#	Vietnamese text	Transcription	English meaning
1	Ở đây có buôn bán dê không?	[ɤ-3 ɗɤ̆j-1 kɔ-5a ɓuən-1 ɓan-5a ɗe-1 xoŋ͡m-1]	Do you sell *goats* here?
2	Ở đây có buôn bán dễ không?	[ɤ-3 ɗɤ̆j-1 kɔ-5a ɓuən-1 ɓan-5a ɗe-4 xoŋ͡m-1]	Do you sell *easily* here?
3	Ở đây có buôn bán dế không?	[ɤ-3 ɗɤ̆j-1 kɔ-5a ɓuən-1 ɓan-5a ɗe-5a xoŋ͡m-1]	Do you sell *crickets* here?
4	Mỗi tối, bác sĩ Thuỷ thường đến hỏi thăm các bệnh nhân	moj-4 toj-5a ɓak-5b si-4 tʰʷi-3 tʰɯəŋ-2 ɗen-5a hɔj-3 tʰăm-1 kak-5b ɓe̯ŋ-6a ɲɤ̆n-1	Every evening, doctor *Thuỷ* usually visits her patients
5	Mỗi tối, bác sĩ Thuỳ thường đến hỏi thăm các bệnh nhân	moj-4 toj-5a ɓak-5b si-4 tʰʷi-2 tʰɯəŋ-2 ɗen-5a hɔj-3 tʰăm-1 kak-5b ɓe̯ŋ-6a ɲɤ̆n-1	Every evening, doctor *Thuỳ* usually visits her patients
6	Mỗi tối, bác sĩ Thuý thường đến hỏi thăm các bệnh nhân	moj-4 toj-5a ɓak-5b si-4 tʰʷi-5a tʰɯəŋ-2 ɗen-5a hɔj-3 tʰăm-1 kak-5b ɓe̯ŋ-6a ɲɤ̆n-1	Every evening, doctor *Thuý* usually visits her patients
7	Mỗi tối, bác sĩ Thuỵ thường đến hỏi thăm các bệnh nhân	moj-4 toj-5a ɓak-5b si-4 tʰʷi-6a tʰɯəŋ-2 ɗen-5a hɔj-3 tʰăm-1 kak-5b ɓe̯ŋ-6a ɲɤ̆n-1	Every evening, doctor *Thuỵ* usually visits her patients
8	Em phải đọc chữ Tiếp thật rõ nhé!	[ɛm-1 faj-3 ɗɔk-6b tɕɯ-4 tiəp-5b tʰɤ̆t-6b zɔ-4 ɲɛ-5a]	You should articulate the syllable *"Tiếp"*!
9	Em phải đọc chữ Tiệp thật rõ nhé!	[ɛm-1 faj-3 ɗɔk-6b tɕɯ-4 tiəp-6b tʰɤ̆t-6b zɔ-4 ɲɛ-5a]	You should articulate the syllable *"Tiệp"*!
10	Thế này mà nhà cô còn tiệc gì nữa?	[tʰe-5a năj-2 ma-2 ɲa-2 ko-1 kɔn-2 tiək-6b zi-2 nɯə-4]	In this situation, why does your family still *regret*?
11	Thế này mà nhà cô còn tiếc gì nữa?	[tʰe-5a năj-2 ma-2 ɲa-2 ko-1 kɔn-2 tiək-5b zi-2 nɯə-4]	In this situation, why does your family still *have a party*?
12	Mỗi tôi đi ra đường sợ gặp ma	[moj-4 toj-1 ɗi-1 za-1 ɗuəŋ-2 sɤ-6a ɣăp-6b ma-1]	*It was only me* who feel scared when going outside.
13	Mỗi tối đi ra đường sợ gặp ma	[moj-4 toj-5a ɗi-1 za-1 ɗuəŋ-2 sɤ-6a ɣăp-6b ma-1]	*Every evening* (someone) feel scared when going outside.
14	Mỗi tội đi ra đường sợ gặp ma	[moj-4 toj-6a ɗi-1 za-1 ɗuəŋ-2 sɤ-6a ɣăp-6b ma-1]	*None the less* (someone) feel scared when going outside.

Table C.2 – Test corpus examples of MOS test

#	Vietnamese text	Transcription	English meaning
1	Có nhiều thứ tôi được hoá thân vào số phận các nhân vật khác nhau	[ko-5a ɲiəw-2 tʰɯ-5a toj-1 faj-3 hʷa-5a tʰɤ̆n-1 vaw-2 kak-5b ɲɤ̆n-1 vɤ̆t-6b xak-5b ɲă̤u-1]	There were many films I could incarnate into different characters
2	Điều này tưởng chừng như đơn giản nhưng không phải dễ nhận ra nhất là khi mình thành công lúc còn quá trẻ	[ɗiəw-2 nă̤j-2 tɯəŋ-3 tɕɯŋ-2 ɲɯ-1 ɗɤn-1 zan-3 ɲɯŋ-1 xoŋ͡m-1 faj-3 ze-4 ɲɤ̆n-6a za-1 ɲɤ̆t-5b la-2 xi-1 miŋ̰-2 kɔn-2 kʷa-5a tɕɛ-3]	This seems to be simple yet not always recognizable especially when we have been successful at a very young age.
3	Kiểu đi lắc lư ngộ nghĩnh của chim cánh cụt chính là một cách thức thông minh để đạt được điều đó	[kiəw-3 ɗi-1 lă̤k-5b lɯ-1 ŋo-6a ŋiŋ̰-4 kuə-3 tɕim-1 kĕŋ̰-5a kut-6b tɕiŋ̰-5a la-2 kĕk-5b tʰɯk-5b tʰoŋ͡m-1 miŋ̰-1 ɗe-3 ɗat-6b ɗuək-6b ɗiəw-2 ɗo-6a]	Penguins' adorable swaying walking style is a smart way to get that.
4	Trách nhiệm đó thuộc về ai?	[tɕĕk-5b ɲiəm-6a ɗɔ-5a tʰuək-6b ve-2 aj-1]	To whom that responsibility belongs?
5	Còn công đoàn thì cho rằng họ đơn giản chỉ phản đối cách làm của giới chủ vốn đang phớt lờ mọi nguyên tắc an toàn trong công việc	[kɔn-2 koŋ͡m-1 ɗʷan-2 tʰi-2 tɕɔ-1 ză̤ŋ-2 hɔ-6a ɗɤn-1 zan-3 tɕi-3 fan-3 ɗoj-5a kĕk-5b lam-2 kuə-3 zɤj-5a tɕu-3 von-5a ɗaŋ-1 fɤt-5b lɤ-2 moj-6a ŋʷiən-1 tă̤k-5b an-1 tʷan-2 tɕɔŋ͡m-1 koŋ͡m-1 viək-6b]	And the unions also argue that they simply oppose the capital owners in ignoring all safety rules at work.
6	Nhà này rộng bao nhiêu mét?	[ɲa-2 nă̤j-2 zoŋ͡m-6a ɓaw-1 ɲiəw-1 mɛt-5b]	How big is this house?
7	Xa quá!	[sa-1 kʷa-5a]	Too far!
8	Thế em đã biết nấu chưa?	[tʰe-5a em-1 ɗa-4 ɓiət-5b nă̤w-5a tɕɯə-1]	Do you know how to cook?
9	Trong những ngày vui này còn có tiết mục mừng tuổi, chúc Tết	[tɕɔŋ͡m-1 ɲɯŋ-4 ŋă̤j-2 vuj-1 nă̤j-2 kɔn-2 kɔ-5a tiət-5b muk-6b mɯŋ-2 tuəj-3 tɕuk-5b tet-5b]	During these happy days, there are also some activities like giving lucky money and wishing best wishes.
10	Lần sau chị mua nữa nhé!	[lɤ̆n-2 să̤w-1 tɕi-6a muə-1 nɯa-4 ɲɛ-5a]	Buy this next time, please!
11	Do vậy, muốn nâng được một trọng lượng nhất định thì chỉ còn cách là làm rộng hết cỡ có thể diện tích cánh máy bay để có được lực nâng cần thiết.	[ɗɔ vɤ̆j-6a muən-5a nɤ̆ŋ ɗuək-6b mot-6b ɪtCOŋ͡m-6a lɯəŋ-6a ɲɤ̆t-5b ɗiŋ̰-6a tʰi-2 tɕi-3 kɔn-2 kĕk-5b la-2 lam-2 zoŋ͡m-6a het-5b kɤ-4 kɔ-5a tᵗe-3 ziən-6a tiḳ kĕŋ̰-5a mă̤j-5a ɓă̤j-1 ɗe-3 kɔ-5a ɗuək-6a lɯk-6b nɤ̆k-6b nɤ̆ŋ-1 kɤ̆n-2 tʰiət-5b]	Therefore, to elevate a certain weight, the only way is to increase the area of the wings as much as possible to get the necessary lifting force.
12	Bởi trước khi rời nước, anh chị đã ký khế ước với quốc gia Việt Nam là sẽ về nước phục vụ sau khi tốt nghiệp.	[ɓɤj-3 tɕuək-5b xi-1 zɤj-2 nuək-5b ĕŋ̰ tɕi-6a ɗa-4 ki-5a xe-5a uək-5b vɤj-5a kʷok͡p-5b za-1 viət-6b nam-1 la-2 sɛ-4 ve-2 nɯək-5a fuk-6b vu-6a să̤w xi-1 tot-5b ŋiəp-6b]	Because before leaving the country, you had signed an agreement with the Vietnamese government that you would serve the country after graduation.

Table C.3 – Test corpus examples of Intelligibility test

#	Vietnamese text	Transcription	English meaning
1	Ông là ếch ngồi đáy giếng.	[oŋ͡m-1 la-2 eḵ-5b ŋoj-2 ɗăj-5a ziəŋ-5a]	You are the frog in the well (see no further than your nose)
2	Loanh quanh một lát, bọ muỗm đã mệt phờ.	[lʷĕŋ-1 [kʷĕŋ-1 mot-6b lat-5b ɓɔ-6a muəm-4 ɗa-4 met-6b fɤ-2]	After flying around for a while, beetles were exhausted.
3	Chúng mọc um tùm và trở thành nơi trú ngụ khá tốt cho loài chim này.	[tɕuŋ͡m-5a mɔk-6b um-1 tum-2 va-2 tɕɤ-3 tʰĕŋ-2 nɤj-1 tɕu-5a ŋu-6a xa-5a tot-5b kuə-3 kak-5b lʷaj-2 tɕim-1 năj-2]	They overgrew and became good shelter for this bird species.
4	Thân nỏ được làm bằng gỗ nghiến với dạng thớ mịn nhưng quánh, dẻo.	[tʰɤ̆n-1 nɔ-3 ɗuək-6a lam-2 ɓăŋ-2 ɣo-4 ŋiən-5a vɤj-5a ɗaŋ-5a tʰɤ-5a min-6a ɲɯŋ-1 [kʷĕŋ-5a ɗɛw-3]	The archery's body was made by garnishing wood in smooth yet consistent and elastic lines.
5	Đội đã liên hệ để đưa hai liệt sĩ này về an táng ở nghĩa trang liệt sĩ quê nhà.	[ɗoj-5a ɗa-4 liən-1 he-5a ɗe-3 ɗuə-1 haj-1 liət-6a si-4 năj-2 ve-2 an-1 taŋ-5a ɤ-3 ŋiə-4 tɕaŋ-1 kʷe-1 ŋa-2]	The team had managed to bring these two martyrs back to their home cemetery.
6	Đây là những yếu tố gây áp lực tăng giá đối với thị trường trong nước.	[ɗɤ̆j-1 la-2 ɲɯŋ-4 iəw-5a to-5a ɣɤ̆j-1 ap-5b luk-6b tăŋ-1 za-1 ɗoj-5a vɤj-5a tʰi-6a tɕɯəŋ-2 tɕɔŋ͡m-1 nɯək-5b]	These are the factors that pressure prices to go upward in the domestic market.
7	Hai mắt Mừng tự dưng nhoè ướt.	[haj-1 măt-5b mɯŋ-2 tɯ-6a ɗɯŋ-1 ɲʷɛ-2 ɯət-5b]	Two eyes of Mung suddenly swept and blurred.
8	Triển lãm trưng bày theo bốn mảng đề tài Cổ vật Phật giáo, kỷ lục Phật giáo, mỹ thuật Phật giáo, ảnh nghệ thuật về Phật giáo.	[tɕiən-3 lam-4 tɕɯŋ-1 ɓăj-2 tʰɛw-1 ɓon-5a maŋ-3 ɗe-2 taj-2 ko-3 vat-5b fat-6a zaw-5a ki-3 luk-6a fat-6b zaw-6a mi-4 tʰʷɤ̆t-5b ĕŋ-3 ŋe-6a ve-2 fat-6b zaw-5a]	The exhibition displays four topics: Buddhist antiques, record Buddhism, Buddhist artwork and Buddhist photography.
9	Nó thò đầu ra - vẻ mặt phớn phở, say bứ bừ nói: "chúng nó phắn hết rồi".	[nɔ-5a tʰɔ-5a ɗɤ̆w-2 za-1 vɛ-3 măt-6b fɤn-5a fɤ-3 săj-1 ɓɯ-5a ɓɯ-2 nɔj-5a]	He popped his head out, ecstatic drunk, he said "They're all gone".
10	Thấy thế, thằng bé bệu bạo khóc, nhe hàm răng đầy bựa mám mồi đầy ngún vào lưỡi câu rồi đưa cho hắn.	[tʰɤ̆j-5a tʰe-5a tʰăŋ-2 ɓɛ-5a ɓew-6a ɓuə-6a mam-5a moj-2 ɗɤ̆j-2 ŋun-5a vaw-2 luəj-4 kɤ̆w zoj-2 ɗuə-1 tɕɔ-1 hăn-5a]	Seeing this, the boy cried and bared the teeth filled with plaque then handed him the hook.
11	Nó dịch cái bụng phề phệ đến chuồng làm bọn heo kêu ìn ịt.	[nɔ-5a ɗiḵ-6b kaj-5a ɓuŋ͡m-6a fe-6a ɗen-5a tɕuəŋ-2 lam-2 ɓɔn-6a hɛw kew-1 in-2 it-6b]	He moved the potbelly to the stable, making the pigs to oink.
12	Diệt xong họ Trịnh, Nguyễn Huệ tới yết kiến vua Hiển Tông.	[ɗiət-6b sɤ̆ŋ-1 hɔ-6a tɕiŋ ŋʷiən-4 hʷe-6a tɤj-5a iət kiən-5a vuə hiən-3 toŋ͡m-1]	After slaying the Trinh families, Nguyen Hue encountered King Hien Tong.

Table C.4 – Test corpus examples of Pair-wise preference test using syntactic rules

#	Vietnamese text	Transcription	English meaning
1	Khách nước ngoài đến Việt Nam ăn Tết thường cảm thấy thú vị	[xăk-5b nɯək-5b ŋʷaj-2 ɗen-5a viət-6b nam-1 ăn-1 tet-5b tʰɯəŋ-2 kam-3 tʰuɤj-5a tʰu-5a vi-6a]	Foreigners who come to Vietnam for the Tet holiday often find it interesting.
2	Những ca khúc của ông mang tính tự sự lắng đọng	[ɲɯŋ-4 ka-1 xukp͡-5b kuə-3 oŋm͡-1 maŋ-1 tiŋ̟-5a tɯ-6a sɯ-5a lăŋ-5a ɗɔŋm͡-6a]	His songs are narrative and sentimental.
3	Tôi nhận ra rằng lúc mình đang ở đỉnh cao không bao giờ thiếu người quan tâm	[toj-1 ɲɤ̆n-6a za-1 zăŋ-2 lukp͡-5b miŋ̟-2 ɤ-3 ɗiŋ̟ kaw-1 xoŋm͡-1 ɓaw-1 zɤ-2 tʰiəw-5a ŋɯəj-2 kʷan-1 tɤ̆m-1]	I realized that when I was at the top, it's never lack of people who were interested in me.
4	Những giao dịch đó có thực hay không thì chưa ai có thể trả lời một cách chính xác	[ɲɯŋ-4 zaw-1 ziḳ̟-6a ɗɔ-5a kɔ-5a tʰɯk-6b hăj-1 xoŋm͡-1 tⁱ-2 tɕɯə-1 aj-1 kɔ-5a tʰe-3 tɕa-3 lɤj-2 mot-6b kuɛḳ̟-5b tɕiŋ̟-5a sak-5b]	No one can really answer whether those transactions are real or not.
5	Cháu đã đeo một chiếc nhẫn rất cứng vào dương vật khiến nó bị thít chặt lại	[tɕăw-5a ɗa-4 ɗɛw-1 mot-6b tɕiək-5b ɲɤ̆n-4 kim-1 kɯəŋ-1 zɤ̆t-5b kɯŋ-5a vaw-2 zɯəŋ-1 vɤ̆t-6b xiən-5a nɔ-5a ɓi-6a tʰit-5b tɕăt-6b laj-5a]	The child wore a very hard ring on his penis causing it to be tightened.
6	Mãnh có cha luôn say rượu đánh đập vợ con	[mɛ̆ŋ̟-4 kɔ-5a tɕa-1 luən-1 săk-1̟ zɯəw-6a ɗɛ̆ŋ̟-5a ɗɤ̆p-6b vɤ-6a kɔn-1]	Manh has a drunken father who always beat his wife and children.
7	Thời tiết xấu máy bay không xuống được	[tʰɤj-2 tiət-5b sɤ̆w-5a măj-5a ɓăj-1 xoŋm͡-1 suəŋ-5a ɗɯək-6a]	The plane could not land because of bad weather.
8	Sự vươn lên và thành công của những người ấy làm tôi nể phục.	[sɯ-6a vɯɤn-1 len-1 va-2 tʰăŋ̟-2 koŋm͡-1 kuə-3 ɲɯŋ-4 ŋɯəj̟-2 ɤ̆j-5a lam-2 toj-1 ne-3 fukp͡-6b]	I have great admiration to the rise and success of those people.
9	Cậu luôn ý thức bảo vệ mẹ mình và thường giúp đỡ bạn bè trong lúc khó khăn	[kɤ̆w-5a luən-1 i-5a tʰɯk-5a ɓaw-3 ve-6a mɛ-6a miŋ̟-2 va-2 tʰɯəŋ-2 zukp͡-5a ɗɤ-4 ɓan-6a ɗɛ-2 tɕɔŋm͡]-1 lukp͡-5b xɔ-5a xăn-1	He always protects his mother and help friends in difficult situations.
10	Tại thành phố Hồ Chí Minh có nhiều đường dây cá độ bóng đá rất chuyên nghiệp.	taj-6a tʰăŋ̟-2 fo-5a ho-2 tɕi-5a miŋ̟-1 kɔ zɤ̆t-5b ɲiəw-2 ɗuəŋ̟-2 zɤ̆j-1 ka-5a ɗo-6a ɓɔŋm͡-5a ɗa-5a zɤ̆t-5b tɕʷiən-1 ŋiəp-6b	In Ho Chi Minh City, there are many professional football gambling services.
11	Điều này gây ra cảm giác đau ở cẳng chân sưng nề ở mắt cá chân.	[ɗiəw-2 năj-2 ɣɤ̆j-1 kam-3 zak-5b ɗăw ɤ-3 kăŋ-3 tɕɤ̆n-1 suŋ-1 ne-2 ɤ-3 măt-5b ka-5a tɕɤ̆n-1]	This causes pain in the legs and swelling in the ankles.
12	Ngoài ra tại đây còn nhận tổ chức liên hoan sinh nhật đám cưới.	[ŋʷaj-2 za-1 taj-6a ɗɤ̆j-1 kɔn-2 ɲɤ̆n-6a liən-1 hʷan-1 siŋ̟ ɲɤ̆t-6b ɗm-5a kuəj-5a]	Besides, they also organize weddings and birthday parties in here.

Table C.5 – Test corpus examples of Pair-wise preference test using syntactic-blocks, -links and POSs

#	Vietnamese text	Transcription	English meaning
1	Lời chúc càng tự nhiên chân thành càng được yêu thích	[lɤj-2 tɕuk-5b kaŋ-2 tW-6a ɲiən-1 tɕɤn-1 t^hĕŋ-2 kaŋ-2 ɗɯək-6a iəj-1 t^hi[k]-5b]	The more natural and sincere the wishes, the more they are favored.
2	Cậu hay ăn hiếp những bạn yếu thế hơn mình và biết cách luồn lách sao cho có lợi về mình	[kў̆w-6a hăj-1 ăn-1 hiəp-5b ɲɯŋ-4 ɓan-6a iəw-5a t^he-5a hɤn-1 miŋ-2 va-2 ɓiət-5b kuɛk̆-5b luən-2 lak-5a saw-1 tɕɔ-1 kɔ-5a lɤj-6a ve-2 miŋ̆-2]	He often bullies the weaker and cheats others to get benefits.
3	Cấm lưu thông tự do thảo quả là chủ trương của nhà nước hay của một ngành nào đó mà dẫn đến nông nỗi này	[kў̆m-5a lɯw-1 t^hoŋ͡m-1 tɯ-6a zɔ-1 t^haw-3 k^wa-3 la-2 tɕu-3 tɕɯəŋ-1 kuə-3 ɲa-2 nɯək-5a hăj-1 kuə-3 mot-6b to-3 tɕɯk-5b naw-2 ɗɔ-5a ma-2 zў̆n-4 noŋ͡m-1 noj-4 năj-2]	Banned free circulation of cardamom is policy of state or other sector that lead to this consequence.
4	Cặp bài trùng này đã cướp trắng của đồng bọn 170kg thuốc phiện	[kăp-5b ɓaj-2 tɕuŋ͡m-2 năj-2 ɗa-4 kɯəp-5b tɕăŋ-5a kuə-3 ɗoŋ͡m ɓɔn-6a mot-6b tɕăm-1 ɓў̆j-3 mɯəj-1 ki-1 lo-1 ɣam-1 t^huək-5b fiən-6b]	This matching pair stole 170 kg drugs from their accomplices.
5	Một lần nữa hồ sơ tội ác của Lượng dầy thêm	[mot-6b lў̆n-2 nɯə-4 ho-2 sɤ-1 kuə-3 lɯəŋ-6a ɗў̆j-2 t^hem-1]	Luong's criminal record increases once more time.
6	Người thứ ba cũng có chút ít lỗi vô ý chính là bạn đấy Hiên ạ	[ŋɯəj-2 t^hɯ-5a ɓa-1 kuŋ͡m-4 kɔ-6a tɕut-5b it-5b loj-4 vo-1 i-5a tɕiŋ̆-5a la-2 ɓan-6a ɗў̆j-6a hiən-1 a-6a]	The third person who is accountable for some unintentional faults is you, Hien.
7	Còn anh?	[kɔn-2 ĕŋ̆-1]	And you?
8	Ông làm ơn chỉ cho chúng tôi trường tiểu học Giảng Võ được không?	[oŋ͡m-1 lam-2 ɤn-1 tɕi-3 tɕɔ-1 tɕuŋ͡m-5a toj-1 tɕɯəŋ-2 tiəw-3 hɔk-6a zaŋ-3 vo-4 ɗɯək-6b xoŋ͡m-1]	Can you show me the way to Giang Vo primary school?
9	Cấm lưu thông tự do thảo quả là chủ trương của nhà nước hay của một ngành nào đó mà dẫn đến nông nỗi này.	[kў̆m-5a lɯw-1 t^hoŋ͡m-1 tɯ-6a ɗɔ-1 t^haw-3 k^wa-3 la-2 tɕu-3 tɕɯəŋ-1 kɯə-3 ɲa-2 nuək-5b hăj-1 kuə-3 mot-5b ŋĕŋ̆-2 naw-2 ɗɔ-5a ma-2 zў̆n-4 ɗen-5a noŋ͡m-1 noj-4 năj-2]	Banned free circulation of cardamom is policy of state or other sector that lead to this consequence.
10	Riêng hai hồ nước lớn sẽ là nơi truyền bá cho môn thể thao mới nhất Việt Nam: đi bộ trên nước.	[ziəŋ-1 haj-1 ho-2 nɯək-5b sɛ-4 la-2 nɤj-1 tɕ^{w}iən-2 ba-5a tɕɔ-1 mon-1 t^he-3 t^haw-1 mɤj-5a ɲў̆t-5b viət-6b nam-1 ɗi-1 ɓo-6a tɕen-1 nɯək-5b]	The two large lakes will be our locations to spread the latest sport in Vietnam: walking on water.
11	Học trò tôi rất đông nhưng mở lớp dậy chính thức thì tôi chưa làm được.	[hɔk͡p-6b tɕɔ-2 toj-1 zў̆t-5b ɗoŋ͡m-1 ɲɯŋ-1 mɤ-3 lɤp-5b ɗў̆j-6a tɕiŋ̆-5a t^hɯk-5b t^hi-2 toj-1 tɕɯə-1 lam-2 ɗɯək-6b]	I have many students but I haven't opened any official classes yet.
12	Anh thử tìm trong cặp hay trong túi xem?	[ĕŋ̆-1 t^hɯ-3 tim-2 tɕoŋ͡m-1 t^{w}i-5a suɛk̆-5b hăj-1 tɕoŋ̆-1 t^{w}i-5a sɛm-1]	Try to look for it in your briefcase and bag!

Bibliography

Abushariah Mohammad A., Ainon Raja N., Zainuddin Roziati, Elshafei Moustafa, and Khalifa Othman O. Phonetically Rich and Balanced Text and Speech Corpora for Arabic Language. *Journal Language Resources and Evaluation*, 46(4):601–634, December 2012. ISSN 1574-020X. doi: 10.1007/s10579-011-9166-8.

Anumanchipalli Gopala Krishna, Prahallad Kishore, and Black Alan W. Festvox: Tools for creation and analyses of large speech corpora. In *Workshop on Very Large Scale Phonetics Research*, UPenn, Philadelphia, 2011.

Apel Jens, Neubarth Friedrich, Pirker Hannes, and Trost Harald. Have a break! Modelling pauses in German speech. In *KONVENS*, pages 5–12. KONVENS, 2004.

Beckman Mary E. and Hirschberg Julia. The ToBI Annotation Conventions. Technical report, Ohio State University, 1994.

Bikel Daniel M. *On the parameter space of generative lexicalized statistical parsing models.* PhD thesis, University of Pennsylvania, United States, 2004.

Brunelle Marc. Tonal coarticulation effects in northern vietnamese. In *Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS 15)*, pages 2673–2676, Barcelona, 2003.

Brunelle Marc. Northern and southern vietnamese tone coarticulation: A comparative case study. *Journal of Southeast Asian Linguistics*, 1:49–62, 2009.

Campbell Nick. Automatic detection of prosodic boundaries in speech. *Speech communication*, 13(3):343–354, 1993.

Campbell W. Nick. Syllable-based segmental duration. *Talking machines: Theories, models, and designs*, pages 211–224, 1992.

Cao Xuân Hạo. Le problème du phonème en vietnamien, 1975.

Carreras Xavier, Collins Michael, and Koo Terry. Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, CoNLL '08, pages 9–16, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-905593-48-4.

Charniak Eugene. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 132–139, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

Charniak Eugene and Johnson Mark. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219862.

Chevelu Jonathan, Barbot Nelly, Boëffard Olivier, and Delhay Arnaud. Comparing Set-Covering Strategies for Optimal Corpus Design. In *Proceedings of the International Conference on Language Resources and Evaluation*, Morocco, 2008.

Chistikov Pavel and Khomitsevich Olga. Improving Prosodic Break Detection in a Russian TTS System. In *Speech and Computer*, pages 181–188. Springer, 2013.

Chomphan Suphattharachai. Towards the Development of Speaker-Dependent and Speaker-Independent Hidden Markov Model-Based Thai Speech Synthesis. *Journal of Computer Science*, 5(12):905–914, December 2009. ISSN 15493636. doi: 10.3844/jcssp.2009.905.914.

Chomphan Suphattharachai. Analysis of Decision Trees in Context Clustering of Hidden Markov Model Based Thai Speech Synthesis. *Journal of Computer Science*, 7(3):359–365, March 2011. ISSN 1549-3636. doi: 10.3844/jcssp.2011.359.365.

Chomphan Suphattharachai and Chompunth Chutarat. Improvement of Tone Intelligibility for Average-Voice-Based Thai Speech Synthesis. *American Journal of Applied Sciences*, 9 (3):358–364, 2012. ISSN 1546-9239.

Chomphan Suphattharachai and Kobayashi Takao. Design of tree-based context clustering for an HMM-based Thai speech synthesis system. In *Proceeding of the 6th ISCA Workshop on Speech Synthesis*, pages 160–165, Bonn, Germany, August 2007.

Chomphan Suphattharachai and Kobayashi Takao. Tone correctness improvement in speaker dependent HMM-based Thai speech synthesis. *Speech Communication*, 50(5):392–404, 2008. ISSN 0167-6393.

Chomphan Suphattharachai and Kobayashi Takao. Tone correctness improvement in speaker-independent average-voice-based Thai speech synthesis. *Speech Communication*, 51(4): 330–343, 2009. ISSN 0167-6393.

Chou Fu-Chiang, Tseng Chiu yu, and Lee Lin-Shan. Automatic generation of prosodic structure for high quality Mandarin speech synthesis. In *ICSLP*. ISCA, 1996.

Chou Fu-Chiang, Tseng Chiu-yu, and Lee Lin-shan. A set of corpus-based text-to-speech synthesis technologies for Mandarin Chinese. *IEEE Transactions on Speech and Audio Processing*, 10(7):481–494, October 2002. ISSN 1063-6676. doi: 10.1109/TSA.2002.803437.

Clark Alexander Simon, Fox Chris, and Lappin Shalom, editors. *The handbook of computational linguistics and natural language processing.* Blackwell handbooks in linguistics. Wiley-Blackwell, Chichester, paperback ed edition, 2013. ISBN 9781405155816 9781118347188 9781405155816.

Cochran William G. and Cox Gertrude M. *Experimental Designs, 2nd Edition.* Wiley, 2 edition, April 1992. ISBN 0471545678.

Collins Michael. *Head-Driven Statistical Models for Natural Language Parsing.* PhD thesis, University of Pennsylvania, 1999.

Collins Michael. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics, July 2002. doi: 10.3115/1118693.1118694.

Collins Michael and Roark Brian. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 111–118, Barcelona, Spain, July 2004. doi: 10.3115/1218955.1218970.

Dao Loan. The Vietnamese classifiers 'CON', 'CÁI' and the Natural Semantic Metalanguage (NSM) approach: A preliminary study. In *Proceedings of the 42nd Australian Linguistic Society Conference*, Australia, 2011.

Dinh Anh-Tuan, Phan Thanh-Son, Vu Tat-Thang, and Luong Chi Mai. Vietnamese hmm-based speech synthesis with prosody information. In *8th ISCA Workshop on Speech Synthesis*, pages 51–54, Barcelona, Spain, August 2013.

Do The Dung, Tran Thien Huong, and Boulakia George. *Intonation in Vietnamese (395-416). In Intonation systems: A Survey of Twenty Languages (Hirst Daniel and di Cristo).* Cambridge University Press, December 1998. ISBN 9780521395502.

Do Thi Ngoc Diep, Le Viet Bac, Bigi Brigitte, Besacier Laurent, and Castelli Eric. Mining a comparable text corpus for a vietnamese - french statistical machine translation system. In *The 4th Workshop on statistical machine translation - EACL 2009*, Athens, Greece, March 2009.

Do Tu Trong and Takara Tomio. Precise tone generation for Vietnamese text-to-speech system. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, volume 1, pages I–504–I–507 vol.1, April 2003. doi: 10.1109/ICASSP.2003.1198828.

Do Tu Trong and Takara Tomio. Vietnamese Text-To-Speech system with precise tone generation. *Acoustical Science and Technology*, 25(5):347–353, 2004. ISSN 1346-3969, 1347-5177.

Do Van Thao, Tran Do Dat, and Nguyen Thi Thu Trang. Non-uniform unit selection in Vietnamese speech synthesis. In *Proceedings of the Second Symposium on Information and Communication Technology (SoICT 2011)*, pages 165–171, Hanoi, Vietnam, 2011. ISBN 978-1-4503-0880-9.

Doan Thien Thuat. *Vietnamese phonetics (in Vietnamese).* Vacational School and University Publisher, 1977.

Doan Xuan Kien. Re-consider a problem of Vietnamese phonetics: Vowels (in Vietnamese). *Hop Luu*, 45, 1999a.

Doan Xuan Kien. Re-consider a problem of Vietnamese phonetics: Syllable structure (in Vietnamese). *Hop Luu*, 48:1–24, 1999b.

Donovan R. E., Ittycheriah A., Franz M., Ramabhadran B., Eide E., Viswanathan M., Bakis R., Hamza W., Picheny M., Gleason P., Rutherfoord T., Cox P., Green D., Janke E., Revelin S., Waast C., Zeller B., Guenther C., and Kunzmann J. The IBM Trainable Speech Synthesis System. In *In Proc ICSLP*, 1998.

Doukhan David, Rilliard Albert, Rosset Sophie, and d'Alessandro Christophe. Modelling pause duration as a function of contextual length. In *INTERSPEECH*. ISCA, 2012.

Doval Boris, Alessandro Christophe, and Henrich Nathalie. The Voice Source as a Causal/Anticausal Linear Filter. In *Voice Quality: Functions, Analysis and Synthesis*, pages 16–20, Geneva, Switzerland, 2003.

Duan Quansheng, Kang Shiyin, Wu Zhiyong, Cai Lianhong, Shuang Zhiwei, and Qin Yong. Comparison of Syllable/Phone HMM Based Mandarin TTS. In *2010 20th International Conference on Pattern Recognition (ICPR)*, pages 4496–4499, August 2010. doi: 10.1109/ICPR.2010.1092.

Dutoit Thierry. *An Introduction to Text-to-speech Synthesis*. Kluwer Academic Publishers, Norwell, MA, USA, 1997. ISBN 0-7923-4498-7.

Dutoit Thierry and Stylianou Yannis. Text-to-speech synthesis. *Handbook of Computational Linguistics*, pages 323–338, 2003.

Emeneau Murray Barnson. *Studies in Vietnamese (Annamese) Grammar*. University of California Press, first edition, 1951.

Ferlus Michel. The origin of tones in viet-muong. *The Eleventh Annual Conference of the Southeast Asian Linguistics Society 2001*, pages 297–313, 2001.

Francois Hélène and Boëffard Olivier. Design of an optimal continuous speech database for text-to-speech synthesis considered as a set covering problem. In *Proceedings of the 7th European Conference on Speech Communication and Technology*, pages 829–832, Denmark, 2001.

Freund Yoav and Schapire Robert E. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, December 1999. ISSN 0885-6125. doi: 10.1023/A:1007662407062.

Gibbon Dafydd, Moore Roger, and Winski Richard, editors. *Handbook of Standards and Resources for Spoken Language Systems*. Mouton De Gruyter, Berlin ; New York, July 1997. ISBN 9783110153668.

Grishman Ralph, Macleod Catherine, and Sterling John. Evaluating Parsing Strategies Using Standardized Parse Files. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 156–161, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics. doi: 10.3115/974499.974528.

Guan Yong, Tian Jilei, Wu Yi-jian, Yamagishi Junichi, and Nurminen Jani. An Unified and Automatic Approach of Mandarin HTS System. In *Proceedings of the 7th Speech Synthesis Workshop (SSW7)*, Kyoto, Japan, 2010.

Han Mieko Shimizu. *Vietnamese vowels*, volume 4. Acoustic Phonetics Research Laboratory, University of Southern California, 1966.

Haudricourt André Georges. La place du vietnamien dans les langues austroasiatiques. *Bulletin de la Société de Linguistique de Paris 49(1)*, pages 122—-128, 1953.

Haudricourt André-Georges. The origin of the peculiarities of the Vietnamese alphabet (Translated by Alexis Michaud). *Mon-Khmer Studies*, 39:89–104, 2010.

Hawley Mark S, Cunningham Sean P, Green Phil D, Enderby Pam, Palmer Rebecca, Sehgal Siddharth, and O'Neill Peggy. A voice-input voice-output communication aid for people with severe speech impairment. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 21(1):23–31, 2013.

He Linyu, Yang Jian, Zuo Libo, and Kui Liping. A trainable Vietnamese speech synthesis system based on HMM. In *Proceedings of the International Conference on Electric Information and Control Engineering (ICEICE)*, pages 3910–3913, Wuhan, China, 2011.

Hiroya Fujisaki Keikichi Hirose. Analysis of voice fundamental frequency contours for declarative sentences of Japanese. *Journal of the Acoustical Society of Japan (E)*, 5(4):233–242, 1984. ISSN 0388-2861. doi: 10.1250/ast.5.233.

Hsia Chi-Chun, Wu Chung-Hsien, and Wu Jung-Yun. Exploiting Prosody Hierarchy and Dynamic Features for Pitch Modeling and Generation in HMM-Based Speech Synthesis. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):1994–2003, November 2010. ISSN 1558-7916. doi: 10.1109/TASL.2010.2040791.

Huang Liang. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL-08: HLT*, pages 586–594, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Huang Liang and Sagae Kenji. Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1077–1086, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Huang Xuedong, Acero Alex, and Hon Hsiao-Wuen. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001. ISBN 0130226165.

Hunt A.J. and Black A.W. Unit selection in a concatenative speech synthesis system using a large speech database. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings*, volume 1, pages 373–376 vol. 1, May 1996. doi: 10.1109/ICASSP.1996.541110.

Jilka Matthias, Möhler Gregor, and Dogil Grzegorz. Rules for the Generation of ToBI-based American English Intonation. *Speech Communication*, 28:83–108, June 1999.

Jokisch Oliver, Kruschke Hans, and Hoffmann Rüdiger. Prosodic reading style simulation for text-to-speech synthesis. In *Affective Computing and Intelligent Interaction*, pages 426–432. Springer, 2005.

Keri Venkatesh, Pammi Sathish Chandra, and Prahallad Kishore. Pause prediction from lexical and syntax information. In *Proceedings of International Conference on Natural Language Processing (ICON)*, 2007.

Kirby James P. Vietnamese (Hanoi Vietnamese). *Journal of the International Phonetic Association*, 41(03):381–392, 2011.

Klein Dan and Manning Christopher D. A* parsing: Fast exact viterbi parse selection. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 40–47, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073461.

Kroeger Paul. *Analyzing Grammar: An Introduction*. Cambridge University Press, May 2005. ISBN 9780521816229.

Kui Liping, Yang Jian, He Bin, and Hu Enxing. An Experimental Study on Vietnamese Speech Synthesis. In *Proceedings of the International Conference on Asian Language Processing (IALP)*, pages 232–235, Penang, Malaysia, 2011.

Lafferty John D., McCallum Andrew, and Pereira Fernando C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

Laver John. *Principles of Phonetics*. Cambridge University Press, 1994. ISBN 9780521456555.

Le Anh-Cuong, Nguyen Phuong-Thai, Vuong Hoai-Thu, Pham Minh-Thu, and Ho Tu-Bao. An experimental study on lexicalized statistical parsing for vietnamese. *Knowledge and Systems Engineering, International Conference on*, 0:162–167, 2009. doi: http://doi.ieeecomputersociety.org/10.1109/KSE.2009.41.

Le Anh-Tu, Tran Do-Dat, and Nguyen Thi Thu Trang. A Model of F0 Contour for Vietnamese Questions, Applied in Speech Synthesis. In *Proceedings of the Second Symposium on Information and Communication Technology (SoICT 2011)*, SoICT '11, pages 172–178, Hanoi, Vietnam, 2011. ACM. ISBN 978-1-4503-0880-9. doi: 10.1145/2069216.2069250.

Le Hong Phuong, Nguyen Thi Minh Huyen, Roussanaly Azim, and Ho Tuong Vinh. *A Hybrid Approach to Word Segmentation of Vietnamese Texts*, volume 5196. Springer-Verlag Berlin, Heidelberg ©2008, 2008. ISBN 978-3-540-88281-7.

Le Hong Phuong, Roussanaly Azim, Nguyen Thi Minh Huyen, and Rossignol Mathias. An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts. In *Traitement Automatique des Langues Naturelles - TALN 2010*, Montreal, Canada, 2010.

Le Hong Phuong, Nguyen Thi Minh Huyen, and Roussanaly Azim. Vietnamese parsing with an automatically extracted tree-adjoining grammar. In *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), Ho Chi Minh City, Vietnam, February 27 - March 1, 2012*, pages 1–6, 2012. doi: 10.1109/rivf.2012.6169832.

Le Quang Thang, Noji Hiroshi, and Miyao Yusuke. Optimal Shift-Reduce Constituent Parsing with Structured Perceptron. In *Proceeding of the 7th International Joint conference on Natural Language Processing of the Association for Computational Linguistics*, Beijing, China, July 2015.

Le Van Ly. *Le parler vietnamien essai d'une grammaire vietnamienne*. PhD thesis, Université de Paris (1896-1968), Paris, 1948.

Le Viet Bac, Tran Do Dat, Castelli Eric, Besacier Laurent, and Serignat Jean-François. Spoken and Written Language Resources for Vietnamese. *LREC*, 4:599–602, 2004.

Le Viet-Bac, Tran Do-Dat, Besacier Laurent, Castelli Eric, and Serignat Jean-François. First steps in building a large vocabulary continuous speech recognition system for Vietnamese. *RIVF 2005. Can Tho, Vietnam*, 2005.

Li Aijun, Pan Shifeng, and Tao Jianhua. HMM-based speech synthesis with a flexible Mandarin stress adaptation model. In *2010 IEEE 10th International Conference on Signal Processing (ICSP)*, pages 625–628, October 2010. doi: 10.1109/ICOSP.2010.5656769.

Li Ya, Tao Jianhua, Hirose Keikichi, Xu Xiaoying, and Lai Wei. Hierarchical stress modeling and generation in mandarin for expressive Text-to-Speech. *Speech Communication*, May 2015. ISSN 0167-6393. doi: 10.1016/j.specom.2015.05.003.

Marcus Mitchell P., Santorini Beatrice, and Marcinkiewicz Mary Ann. Building a large annotated corpus of english: The penn treebank. *COMPUTATIONAL LINGUISTICS*, 19 (2):313–330, 1993.

Martin Philippe. Intonation: A case for experimental phonology. *Les Cahiers de l'ICP. Bulletin de la communication parlée*, (5):89–107, 2000.

Martin Philippe. WinPitch corpus, a text to speech alignment tool for multimodal corpora. In *LREC*, 2004.

Martin Philippe. WinPitch LTL, un logiciel multimédia d'enseignement de la prosodie. *Alsic. Apprentissage des Langues et Systèmes d'Information et de Communication*, 8(2), 2005.

Martin Philippe. Prosodic structure revisited: a cognitive approach. the example of french. In *Speech Prosody 2010*, Chicago, 2010.

Maspero Henri. Études sur la phonétique historique de la langue annamite: Les initiales. *Bulletin de l'École Francaise d'Extrême Orient 12*, pages 114—-116, 1912.

Masuko Takashi. *HMM-Based Speech Synthesis and Its Applications*. PhD thesis, Institute of Technology, Japan, 2002.

McClosky David, Charniak Eugene, and Johnson Mark. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 337–344, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220218.

Michaud Alexis. Final consonants and glottalization: new perspectives from Hanoi Vietnamese. *Phonetica*, 61(2-3):119–146, 2004. ISSN 0031-8388.

Michaud Alexis, Vu-Ngoc Tuân, Amelot Angélique, and Roubeau Bernard. Nasal release, nasal finals and tonal contrasts in Hanoi Vietnamese: an aerodynamic experiment. *Mon-Khmer Studies*, 36:pp. 121–137, 2006.

Michaud Alexis, Ferlus Michel, and Nguyen Minh-Chau. Strata of standardization: the phong nha dialect of vietnamese (quang binh province) in historical perspective. *Linguistics of the Tibeto-Burman Area, Volume 38.1*, April 2015.

Moungsri D., Koriyama T., and Kobayashi T. HMM-based Thai speech synthesis using unsupervised stress context labeling. In *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, pages 1–4, December 2014. doi: 10.1109/APSIPA.2014.7041599.

Navarro Gonzalo. A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, 33:31–88, 2001.

Nespor Marina and Vogel Irene. Prosodic Structure Above the Word. In Cutler Dr Anne and Ladd Dr D. Robert, editors, *Prosody: Models and Measurements*, number 14 in Springer Series in Language and Communication, pages 123–140. Springer Berlin Heidelberg, January 1983. ISBN 978-3-642-69105-8, 978-3-642-69103-4.

Nespor Marina and Vogel Irene. *Prosodic phonology: with a new foreword*, volume 28. Walter de Gruyter, 2007.

Nguyen Dung Tien, Luong Chi Mai, Vu Bang Kim, Mixdorff Hansjoerg, and Ngo Huy Hoang. Fujisaki model based F0 contours in vietnamese TTS. In *INTERSPEECH*. Citeseer, 2004.

Nguyen Huu Quynh. *Vietnamese Grammar (in Vietnamese)*. Bach Khoa Dictionary Publisher, 2007.

Nguyen Phuong-Thai, Vu Xuan-Luong, Nguyen Thi-Minh-Huyen, Nguyen Van-Hiep, and Le Hong-Phuong. Building a Large Syntactically-annotated Corpus of Vietnamese. In *Proceedings of the Third Linguistic Annotation Workshop*, ACL-IJCNLP '09, pages 182–185, Suntec, Singapore, 2009. Association for Computational Linguistics. ISBN 978-1-932432-52-7.

Nguyen Quy, Nguyen Ngan, and Miyao Yusuke. *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, chapter Utilizing State-of-the-art Parsers to Diagnose Problems in Treebank Annotation for a Less Resourced Language, pages 19–27. Association for Computational Linguistics, 2013a.

Nguyen Thi Thu Trang. Demo-voices of VTed Text-To-Speech system. [Online]. Available: https://perso.limsi.fr/trangntt/demo-voices/, 2015a.

Nguyen Thi Thu Trang. PhD web page: HMM-based Vietnamese Text-To-Speech: Prosodic phrasing modeling, corpus design, system design and evaluation. [Online]. Available: https://perso.limsi.fr/trangntt, 2015b.

Nguyen Thi Thu Trang, Pham Thanh Thi, and Tran Do-Dat. A method for Vietnamese text normalization to improve the quality of speech synthesis. In *Proceedings of the First Symposium on Information and Communication Technology (SoICT 2010)*, SoICT '10, pages 78–85, Hanoi, Vietnam, 2010. ACM. ISBN 978-1-4503-0105-3. doi: 10.1145/1852611.1852627.

Nguyen Thi Thu Trang, Alessandro Christophe, Rilliard Albert, and Tran Do Dat. HMM-based TTS for Hanoi Vietnamese: issues in design and evaluation. In *14th Annual Conference of the International Speech Communication Association (Interspeech 2013)*, pages 2311–2315, Lyon, France, August 2013b. ISCA.

Nguyen Thi Thu Trang, Rilliard Albert, Tran Do Dat, and d'Alessandro Christophe. Prosodic phrasing modeling for vietnamese TTS using syntactic information. In *15th Annual Conference of the International Speech Communication Association (Interspeech 2014)*, pages 2332–2336, Singapore, September 2014a. ISCA.

Nguyen Thi Thu Trang, Tran Do Dat, Rilliard Albert, Alessandro Christophe, and Pham Thi Ngoc Yen. Intonation issues in HMM-based speech synthesis for Vietnamese. In *The 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU'14)*, pages 98–104, St. Petersburg, Russia, May 2014b.

Nguyen Thu and Ingram John. Perception of prominence pattern in Vietnamese dissyllabic words. *The Mon-Khmer Studies Journal*, 42, 2013.

Nguyen Van Loi and Edmondson Jerold A. Tones and voice quality in modern northern Vietnamese: Instrumental case studies. *Mon-Khmer Studies Journal*, 28:1–18, 1998.

Oh Yoo Rhee, Kim Yong Guk, Kim Mina, Kim Hong Kook, Lee Mi Suk, and Bae Hyun Joo. Phonetically Balanced Text Corpus Design Using a Similarity Measure for a Stereo Super-Wideband Speech Database. *IEICE Transactions on Information and Systems*, E94-D(7): 1459–1466, July 2011. ISSN 1745-1361, 0916-8532.

Pammi Sathish, Charfuelan Marcela, and Schröder Marc. Multilingual Voice Creation Toolkit for the MARY TTS Platform. In *Language Resources and Evaluation (LREC)*, Malta, 2010.

Parlikar Alok. *Style-Specific Phrasing in Speech Synthesis*. PhD thesis, Carnegie Mellon University, 2013.

Petrov Slav and Klein Dan. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April 2007. Association for Computational Linguistics.

Phan Son Thanh, Vu Thang Tat, and Luong Mai Chi. Extracting MFCC, F0 feature in Vietnamese HMMbased speech synthesis. *International Journal of Electronics and Computer Science Engineering*, 46, 2013a. ISSN 2277-1956.

Phan Thanh Son, Vu Tat Thang, Duong Tu Cuong, and Luong Chi Mai. A study in Vietnamese statistical parametric speech synthesis base on HMM. *International Journal of Advances in Computer Science and Technology*, 2:1–6, 2012.

Phan Thanh-Son, Duong Tu-Cuong, Dinh Anh-Tuan, Vu Tat-Thang, and Luong Chi-Mai. Improvement of naturalness for an HMM-based Vietnamese speech synthesis using the prosodic information. In *2013 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pages 276–281, November 2013b. doi: 10.1109/RIVF.2013.6719907.

Phan Thanh Son, Dinh Anh Tuan, Vu Tat Thang, and Luong Chi Mai. An Improvement of Prosodic Characteristics in Vietnamese Text to Speech System. In Huynh Van Nam, Denoeux Thierry, Tran Dang Hung, Le Anh Cuong, and Pham Son Bao, editors, *Knowledge and Systems Engineering*, number 244 in Advances in Intelligent Systems and Computing, pages 99–111. Springer International Publishing, 2014. ISBN 978-3-319-02740-1, 978-3-319-02741-8.

Powers David M. W. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Technical Report SIE-07-001, School of Informatics and Engineering, Flinders University, Adelaide, Australia, 2007.

Qian Yao and Soong Frank Kao-PingK. HMM-based bilingual (Mandarin-English) TTS techniques, August 2012. U.S. Classification 704/256.3, 704/250, 704/257, 704/256, 704/260, 704/261, 704/258, 704/243; International Classification G10L13/08, G10L15/00, G10L17/00, G10L13/00, G10L15/14; Cooperative Classification G10L13/06; European Classification G10L13/06.

Qian Yao, Soong Frank, Chen Yining, and Chu Min. An HMM-Based Mandarin Chinese Text-To-Speech System. In Huo Qiang, Ma Bin, Chng Eng-Siong, and Li Haizhou, editors, *Chinese Spoken Language Processing*, number 4274 in Lecture Notes in Computer Science, pages 223–232. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-49665-6, 978-3-540-49666-3.

Qian Yao, Cao Houwei, and Soong F.K. HMM-Based Mixed-Language (Mandarin-English) Speech Synthesis. In *6th International Symposium on Chinese Spoken Language Processing, 2008. ISCSLP '08*, pages 1–4, December 2008. doi: 10.1109/CHINSL.2008.ECP.15.

Quinlan J. Ross. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

Rosenblatt F. The perception: A probabilistic model for information storage and organization in the brain. In Anderson James A. and Rosenfeld Edward, editors, *Neurocomputing: Foundations of Research*, pages 89–114. MIT Press, Cambridge, MA, USA, 1988. ISBN 0-262-01097-6.

Sag Ivan A., Wasow Thomas, and Bender Emily M. *Syntactic Theory: A Formal Introduction*. Center for the Study of Language and Information, 2003. ISBN 9781575863993.

Sagae Kenji and Lavie Alon. *Proceedings of the Ninth International Workshop on Parsing Technology*, chapter A Classifier-Based Parser with Linear Run-Time Complexity, pages 125–132. Association for Computational Linguistics, 2005.

Sagae Kenji and Lavie Alon. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 691–698, Sydney, Australia, July 2006. Association for Computational Linguistics.

Sarkar Parakrant and Sreenivasa Rao K. Data-driven pause prediction for speech synthesis in storytelling style speech. In *Communications (NCC), 2015 Twenty First National Conference on*, pages 1–5. IEEE, 2015.

Schröder Marc, Charfuelan Marcela, Pammi Sathish, and Türk Oytun. The MARY TTS entry in the Blizzard Challenge 2008. In *Blizzard Challenge 2008*, Queensland, Australia, 2008.

Selkirk Elisabeth. The Syntax-Phonology Interface. In Goldsmith John, Riggle Jason, and Yu Alan C. L., editors, *The Handbook of Phonological Theory*, pages 435–484. Wiley-Blackwell, 2011. ISBN 9781444343069.

Selkirk Elisabeth O. *On prosodic structure and its relation to syntactic structure*. Indiana University Linguistics Club, 1980.

Shih Chilin and Kochanski Greg. Chinese tone modeling with stem-ml. In *INTERSPEECH*, pages 67–70, 2000.

Silverman Kim, Beckman Mary, and Pierrehumbert . TOBI: A standard scheme for labeling prosody. In *Proceedings of the Second International Conference on Spoken Language Processing*, 1992.

Socher Richard, Bauer John, Manning Christopher D., and Andrew Y. Ng. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Soukoreff R. William and MacKenzie I. Scott. Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic. In *Proceedings of the CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, pages 319–320, New York, USA, 2001. ACM. ISBN 1-58113-340-5.

Tao Jianhua, Dong Honghui, and Zhao Sheng. Rule learning based Chinese prosodic phrase prediction. In *2003 International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings*, pages 425–432, October 2003. doi: 10.1109/NLPKE.2003.1275944.

Tao Jianhua, Liu Fangzhou, Zhang Meng, and Jia Huibin. Design of Speech Corpus for Mandarin Text to Speech. In *The Blizzard Challenge 2008 workshop*, 2008.

Taylor Paul. *Text-to-Speech Synthesis*. Cambridge University Press, Cambridge, UK ; New York, 1 edition edition, March 2009. ISBN 9780521899277.

Thompson Laurence C. *A Vietnamese Reference Grammar*. University of Hawaii Press, 1987. ISBN 9780824811174.

Tokuda K., Zen Heiga, and Black A.W. An HMM-based speech synthesis system applied to English. In *Proceedings of the 2002 IEEE Workshop on Speech Synthesis*, pages 227–230, California, USA, 2002.

Tran Ba Thien. Handbook on accessing information technology for the Blinds. Technical report, Van-Lang University, Ho Chi Minh, Vietnam, February 2007a.

Tran Ba Thien. Survey on requirements of the Blinds to Voice of Southern Vietnam. Technical report, Van-Lang University, Ho Chi Minh, Vietnam, April 2013.

Tran Do Dat. *Synthèse de la parole à partir du texte en langue vietnamienne*. PhD thesis, Grenoble, INPG, January 2007b.

Tran Do Dat and Castelli Eric. Generation of F0 contours for Vietnamese speech synthesis. In *Communications and Electronics (ICCE), 2010 Third International Conference on*, pages 158–162. IEEE, 2010.

Tran Do Dat, Castelli Eric, Serignat Jean-François, Trinh Van Loan, and Lê Xuan Hung. Influence of F0 on Vietnamese Syllable Perception. In *9th European Conference on Speech Communication and Technology (Interspeech 2005)*, pages 1697–1700, 2005.

Tran Do-Dat, Castelli Eric, Serignat Jean-François, and Le Viet-Bac. Analysis and Modeling of Syllable Duration for Vietnamese Speech Synthesis. In *O-COSCODA2007*, 2007.

Uraga Esmeralda and Gamboa César. VOXMEX Speech Database: design of a phonetically balanced corpus. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, volume 46, pages 1–34, Portugal, 2004. doi: 10.1007/s10579-011-9166-8.

Valin Robert D. Van. *An Introduction to Syntax*. Cambridge University Press, April 2001. ISBN 9780521635660.

Villaseñor-Pineda Luis, Gómez Manuel Montes-y, Vaufreydaz Dominique, and Serignat Jean-François. Experiments on the Construction of a Phonetically Balanced Corpus from the

Web. In Gelbukh Alexander, editor, *Computational Linguistics and Intelligent Text Processing*, number 2945 in Lecture Notes in Computer Science, pages 416–419. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-21006-1, 978-3-540-24630-5.

Vogel Irene, Tseng I-Ju Elanna, and Yap Ngee-Thai. Syllable structure in Vietnamese. In *Proceedings of the second Theoretical East Asian Linguistic (TEAL) Workshop*, Taiwan, 2004.

Vu Hai Quan and Cao Xuan Nam. Phrase-based concatenation for Vietnamese TTS (in Vietnamese). *Journal on Information, Technologies, anh Communications (Vietnamese)*, V-1(Projects on research, development and application of Information Technology), 2010.

Vu Minh Quang, Trân Đô Đat, and Castelli Eric. Prosody of interrogative and affirmative sentences in vietnamese language: Analysis and perceptive results. In *Ninth International Conference on Spoken Language Processing*, 2006.

Vu Ngoc Thang and Schultz Tanja. Vietnamese large vocabulary continuous speech recognition. In *IEEE Workshop on Automatic Speech Recognition Understanding, 2009. ASRU 2009*, pages 333–338, November 2009. doi: 10.1109/ASRU.2009.5373424.

Vu Ngoc Thang and Schultz Tanja. Optimization on Vietnamese Large Vocabulary Speech Recognition. In *The 2nd International Workshop on Spoken Language Technologies for Under-resourced Languages*, Malaysia, 2010.

Vu Tat Thang, Nguyen Tien Dung, and Luong Chi Mai. Vietnamese large vocabulary continuous speech recognition. In *Proceeding of 9th European Conference on Speech Communication and Technology*, pages 1689–1692, Portugal, September 2005.

Vu Thang Tat, Luong Mai Chi, and Nakamura S. An HMM-based Vietnamese speech synthesis system. In *Proceedings of the Oriental COCOSDA International Conference on Speech Database and Assessments*, pages 116–121, Beijing, China, 2009.

Wu Zhizheng, Valentini-Botinhao Cassia, Watts Oliver, and King Simon. Deep Neural Networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *Proceedings of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Australia, 2015.

Würgler Simon. Implementation and evaluation of an hmm-based speech generation component for the svox tts system. Master's thesis, Swiss Federal Institute of Technology Zurich, 2011.

Yoshimura Takayoshi. *Simultaneous Modeling of Phonetic and Prosodic Parameters, and Characteristic Conversion for HMM-Based Text-To-Speech Systems.* PhD thesis, Nagoya Institute of Technology, Japan, 2002.

Yoshimura Takayoshi, Tokuda Keiichi, Kobayashi Takao, Masuko Takashi, and Kitamura Tadashi. Simultaneous Modeling Of Spectrum, Pitch And Duration In HMM-Based Speech Synthesis. In *Sixth European Conference on Speech Communication and Technology*, Budapest, Hungary, September 1999.

Yoshimura Takayoshi, Tokuda Keiichi, Masuko Takashi, Kobayashi Takao, and Kitamura Tadashi. Incorporating a mixed excitation model and postfilter into hmm-based text-to-speech synthesis. *Systems and Computers in Japan*, 36(12):43–50, 2005.

Yu Yansuo, Li Dongchen, and Wu Xihong. Prosodic modeling with rich syntactic context in HMM-based Mandarin speech synthesis. In *2013 IEEE China Summit International Conference on Signal and Information Processing (ChinaSIP)*, pages 132–136, July 2013. doi: 10.1109/ChinaSIP.2013.6625313.

Yuan Jiahong, Shih Chilin, and Kochanski Greg P. Comparison of declarative and interrogative intonation in chinese. In *Speech Prosody 2002, an International Conference*, France, 2002.

Zen Heiga, Nose Takashi, Yamagishi Junichi, Sako Shinji, Masuko Takashi, Black Alan, and Tokuda Keiichi. The HMM-based speech synthesis system (HTS) version 2.0. In *6th ISCA Workshop on Speech Synthesis*, pages 294–299, Bonn, Germany, 2007. ISCA.

Zen Heiga, Tokuda Keiichi, and Black Alan W. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064, 2009. ISSN 0167-6393.

Zen Heiga, Senior Andrew, and Schuster Mike. Statistical parametric speech synthesis using Deep Neural Network. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7962–7966, Canada, 2013.

Zeng Xiao-Li, Martin Philippe, and Boulakia Georges. Tones and intonation in declarative and interrogative sentences in mandarin. In *International Symposium on Tonal Aspects of Languages: With Emphasis on Tone Languages*, Beijing, China, 2004.

Zhang Yue and Clark Stephen. Transition-based parsing of the chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 162–171, Paris, France, October 2009. Association for Computational Linguistics.

Zhiwei Shuang Shiyin Kang. Syllable HMM based Mandarin TTS and comparison with concatenative TTS. In *20th IAPR International Conference on Pattern Recognition*, pages 1767–1770, Turkey, August 2010.

Zhu Muhua, Zhang Yue, Chen Wenliang, Zhang Min, and Zhu Jingbo. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 434–443, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Zhu Weibin, Zhang Wei, Shi Qin, Chen Fangxin, Li Haiping, Ma Xijun, and Shen Liqin. Corpus building for data-driven TTS systems. In *Proceedings of 2002 IEEE Workshop on Speech Synthesis, 2002*, pages 199–202, September 2002. doi: 10.1109/WSS.2002.1224408.

UNIVERSITÉ PARIS SUD

Limsi

NGUYEN Thi Thu Trang

HMM-based Vietnamese Text-To-Speech: Prosodic Phrasing Modeling, Corpus Design, System Design, and Evaluation

Abstract

Keywords: speech synthesis, text-to-speech, vietnamese, tonal language, prosodic phrasing modeling.

The thesis objective is to design and build a high quality Hidden Markov Model (HMM-)based Text-To-Speech (TTS) system for Vietnamese – a tonal language. The system is called VTED (Vietnamese TExt-to-speech Development system). In view of the great importance of lexical tones, a "tonophone" – an allophone in tonal context – was proposed as a new speech unit in our TTS system. A new training corpus, VDTS (Vietnamese Di-Tonophone Speech corpus), was designed for 100% coverage of di-phones in tonal contexts (i.e. di-tonophones) using the greedy algorithm from a huge raw text. A total of about 4,000 sentences of VDTS were recorded and pre-processed as a training corpus of VTED.

In the HMM-based speech synthesis, although pause duration can be modeled as a phoneme, the appearance of pauses cannot be predicted by HMMs. Lower phrasing levels above words may not be completely modeled with basic features. This research aimed at automatic prosodic phrasing for Vietnamese TTS using durational clues alone as it appeared too difficult to disentangle intonation from lexical tones. Syntactic blocks, i.e. syntactic phrases with a bounded number of syllables (n), were proposed for predicting final lengthening ($n = 6$) and pause appearance ($n = 10$). Improvements for final lengthening were done by some strategies of grouping single syntactic blocks. The quality of the predictive J48-decision-tree model for pause appearance using syntactic blocks combining with syntactic link and POS (Part-Of-Speech) features reached F-score of 81.4% (Precision=87.6%, Recall=75.9%), much better than that of the model with only POS (F-score=43.6%) or syntactic link (F-score=52.6%) alone.

The architecture of the system was proposed on the basis of the core architecture of HTS with an extension of a Natural Language Processing part for Vietnamese. Pause appearance was predicted by the proposed model. Contextual feature set included phone identity features, locational features, tone-related features, and prosodic features (i.e. POS, final lengthening, break levels). Mary TTS was chosen as a platform for implementing VTED. In the MOS (Mean Opinion Score) test, the first VTED, trained with the old corpus and basic features, was rather good, 0.81 (on a 5 point MOS scale) higher than the previous system – HoaSung (using the non-uniform unit selection with the same training corpus); but still 1.2-1.5 point lower than the natural speech. The quality of the final VTED, trained with the new corpus and prosodic phrasing model, progressed by about 1.04 compared to the first VTED, and its gap with the natural speech was much lessened. In the tone intelligibility test, the final VTED received a high correct rate of 95.4%, only 2.6% lower than the natural speech, and 18% higher than the initial one. The error rate of the first VTED in the intelligibility test with the Latin square design was about 6-12% higher than the natural speech depending on syllable, tone or phone levels. The final one diverged about only 0.4-1.4% from the natural speech.

Résumé

Mots-clefs : synthèse de la parole, text-to-speech, Vietnamien, langue tonale, modélisation de phrasé prosodique.

L'objectif de cette thèse est de concevoir et de construire, un système Text-To-Speech (TTS) haute qualité à base de HMM (Hidden Markov Model) pour le vietnamien, une langue tonale. Le système est appelé VTED (Vietnamese TExt-to-speech Development system). Au vu de la grande importance de tons lexicaux, un "tonophone" – un allophones dans un contexte tonal – a été proposé comme nouvelle unité de la parole dans notre système de TTS. Un nouveau corpus d'entraînement, VDTS (Vietnamese Di-Tonophone Speech corpus), a été conçu à partir d'un grand texte brut pour une couverture de 100% de di-phones tonalisés (di-tonophones) en utilisant l'algorithme glouton. Un total d'environ 4000 phrases ont été enregistrées et pré-traitées comme corpus d'apprentissage de VTED.

Dans la synthèse de la parole sur la base de HMM, bien que la durée de pause puisse être modélisée comme un phonème, l'apparition de pauses ne peut pas être prédite par HMM. Les niveaux de phrasé ne peuvent pas être complètement modélisés avec des caractéristiques de base. Cette recherche vise à obtenir un découpage automatique en groupes intonatifs au moyen des seuls indices de durée. Des blocs syntaxiques constitués de phrases syntaxiques avec un nombre borné de syllabes (n), ont été proposés pour prévoir allongement final ($n = 6$) et pause apparente ($n = 10$). Des améliorations pour allongement final ont été effectuées par des stratégies de regroupement des blocs syntaxiques simples. La qualité du modèle prédictive J48-arbre-décision pour l'apparence de pause à l'aide de blocs syntaxiques, combinée avec lien syntaxique et POS (Part-Of-Speech) dispose atteint un F-score de 81,4 % (Précision = 87,6 %, Recall = 75,9 %), beaucoup mieux que le modèle avec seulement POS (F-score=43,6%) ou un lien syntaxique (F-score=52,6%).

L'architecture du système a été proposée sur la base de l'architecture HTS avec une extension d'une partie traitement du langage naturel pour le Vietnamien. L'apparence de pause a été prédit par le modèle proposé. Les caractéristiques contextuelles incluent les caractéristiques d'identité de "tonophones", les caractéristiques de localisation, les caractéristiques liées à la tonalité, et les caractéristiques prosodiques (POS, allongement final, niveaux de rupture). Mary TTS a été choisi comme plateforme pour la mise en œuvre de VTED. Dans le test MOS (Mean Opinion Score), le premier VTED, appris avec les anciens corpus et des fonctions de base, était plutôt bonne, 0,81 (sur une échelle MOS 5 points) plus élevé que le précédent système – HoaSung (lequel utilise la sélection de l'unité non-uniforme avec le même corpus) ; mais toujours 1,2-1,5 point de moins que le discours naturel. La qualité finale de VTED, avec le nouveau corpus et le modèle de phrasé prosodique, progresse d'environ 1,04 par rapport au premier VTED, et son écart avec le langage naturel a été nettement réduit. Dans le test d'intelligibilité, le VTED final a reçu un bon taux élevé de 95,4%, seulement 2,6% de moins que le discours naturel, et 18% plus élevé que le premier. Le taux d'erreur du premier VTED dans le test d'intelligibilité générale avec le carré latin test d'environ 6-12% plus élevé que le langage naturel selon des niveaux de syllabe, de ton ou par phonème. Le résultat final ne s'écarte de la parole naturelle que de 0,4-1,4%.