VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

IMPLEMENTING AND IMPROVING A SPEECH SYNTHESIS SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. TOMÁŠ BENĚK
AUTHOR

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

IMPLEMENTACE A VYLEPŠENÍ SYSTÉMU SYNTÉZY ŘEČI

IMPLEMENTING AND IMPROVING A SPEECH SYNTHESIS SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. TOMÁŠ BENĚK
AUTHOR

VEDOUCÍ PRÁCE Dipl. Ing. MIRKO HANNEMANN
SUPERVISOR

BRNO 2014

Abstrakt

Tato práce se zabývá syntézou řeči z textu. V práci je podán základní teoretický úvod do syntézy řeči z textu. Práce je postavena na MARY TTS systému, který umožňuje využít existujících modulů k vytvoření vlastního systému pro syntézu řeči z textu, a syntéze řeči pomocí skrytých Markovových modelů natrénovaných na vytvořené řečové databázi. Bylo vytvořeno několik jednoduchých programů ulehčujících vytvoření databáze a přidání nového jazyka a hlasu pro MARY TTS systém bylo demonstrováno. Byl vytvořen a publikován modul a hlas pro Český jazyk. Byl popsán a implementován algoritmus pro přepis grafémů na fonémy.

Abstract

This work deals with text-to-speech synthesis. A general theoretical introduction to TTS is given. This work is based on the MARY TTS system which allows to use existing modules for the creation of an own text-to-speech system and a speech synthesis model using hidden Markov models trained on the created speech database. Several simple programs to ease database creation were created and adding a new language and voice to the MARY TTS system was shown hot to add. The Czech language module and voice for the MARY TTS system was created and published. An algorithm for grapheme-to-phoneme transcription was described and implemented.

Klíčová slova

syntéza řeči, text-to-speech(TTS), systém Mary TTS, HMM syntéza, HTK, HTS, grapheme-to-phoneme transcription(GTP), letter-to-sound(LTS)

Keywords

speech synthesis, text-to-speech(TTS), Mary TTS system, HMM synthesis, HTK, HTS, grapheme-to-phoneme transcription(GTP), letter-to-sound(LTS)

Citace

Tomáš Beněk: Implementing and Improving a Speech Synthesis System, diplomová práce, Brno, FIT VUT v Brně, 2014

Implementing and Improving a Speech Synthesis System

Prohlášení

Prohlašuji, že jsem tuto diplomovou prácí vypracoval samostatně pod vedením pana Dipl. Ing. M. Hannemanna. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

....
Tomáš Beněk
May 28, 2014

Poděkování

Chtěl bych poděkovat vedoucímu práce panu Dipl. Ing. M. Hannemannovi za pomoc a odborné rady při řešení této práce.
Další poděkování patří panu Ing. K. Chalupníčkovi za poskytnutí slovníku obsahujícího fonetickou transkripci českých slov.

Contents

Chapter 1

Introduction

Speech is a natural way of communication between people. It allows to get information from their surroundings without tasking of other senses, mainly vision. Therefore it is desirable to allow this communication channel between human and computer. A speech synthesis allows this. Speech synthesis systems possible applications could be aid for handicaped people (the most familiar is the british physicist S. Hawking), automatic text reading (e.g. SMS in mobile phones) or creation of dialog systems.

This work deals with a text-to-speech synthesis(TTS). It is the most general task of the speech synthesis starting with a plain text and resulting in a synthesized speech signal. The goal of this work is to get familiar with the MARY TTS system, to record a small speech database, to train HMM models for the speech synthesis, to create a demonstrator program to show the system functionality and to concentrate on a selected TTS system component and implement it.

The second chapter of this work provides introduction to a text-to-speech system and its components, the third chapter describes a MARY TTS system which is used as a basis for further work. The chapter four describes adding a new language and a new voice for the MARY TTS system and the chapter five shows solution of a grapheme-to-phoneme transcription problem. The last chapter summarizes the work done and results and suggests further development of the project.

Chapter 2

Text-to-speech synthesis

This chapter describes text-to-speech synthesis and the techniques used in it. Definitions and description are taken from[19], [24], [29] and [30].

The conversion of the text to the speech is the most general and the most difficult task in computer speech synthesis. The result should be the ability of the system to automatically convert arbitrary plain written text to speech which should have a quality equal to speech uttered by a human with good pronunciation.

A typical text-to-speech system consists of two main parts: the natural language processing (NLP) module and the speech synthesis module. The system diagram is depicted in the figure 2.1.

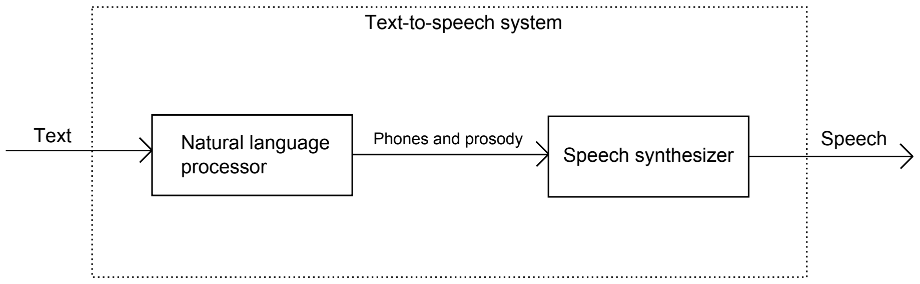


Figure 2.1: TTS system diagram[19]

2.1 Natural language processing

The task of the NLP module is to process the input text, analyse it and gather information used later for the synthesis. This module performs a phonetic transcription and its output is usually a sequence of phones supplied with prosody markers. The NLP module consists of two parts: a phonetic transcription module (grapheme-to-phoneme, GTP) and a prosody generator (PG). In order to increase quality of the resulting speech and eliminating ambiguity of the phonetic transcription and prosody generation, morpho-syntactic analyser (MSA) is also supplied in the NLP module. The NLP module diagram is depicted in the figure 2.2.

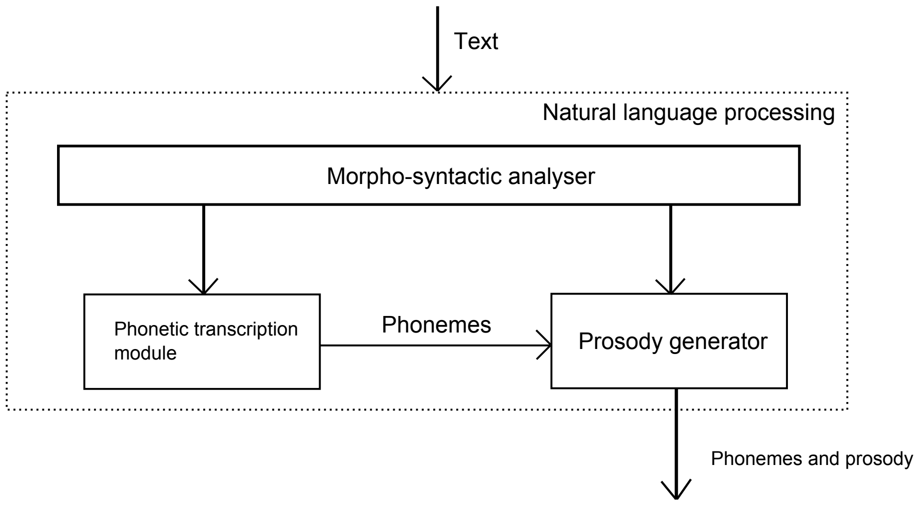

Text
Natural language processing
Morpho-syntactic analyser
Phonetic transcription module
Phonemes
Prosody generator
Phonemes and prosody

Figure 2.2: NLP module diagram[19]

2.1.1 Morpho-syntactic analyser

The MSA module performs the analysis of sentences and represents a sentence as a sequence of tags for grammatical categories of individual words. These categories determine the relations between words and this information is used to determine the exact phonetic transcription. Also prosody generation depends on the sentence syntax. The MSA diagram is depicted in figure 2.3. The individual components are described below.

Text preprocessor

The text preprocessor module performs text unification - it detects the input text type (e.g. plain text, XML) and filters text characters (eliminates white characters, redundant formatting characters and so on). It can also detect text structure like paragraphs or sentences. Sentence end detection is a non-trivial task because not every dot represents sentence end. It can also mark abbreviations or ordinal numbers.

Text normalization

The text normalization module performs the transformation of the input text toa fully verbal form. It converts numbers, abbreviations and acronyms to full text form. There are some problems to solve, for example digits can represent numbers, ordinal numbers, dates, etc. and each one have different form. The text normalization is usually performed using rules and a dictionary.

Morphological analysis

The morphological analysis module proposes the possible part of speech categories for each individual word, on the basis of their spelling. It can divide words according to their lexical category into open set words and closed set words. Closed set words (e.g. prepositions, conjunctions, etc.) can be stored in a dictionary. Open set words can appear in different forms in the text with different prepositions or endings attached and during the analysis

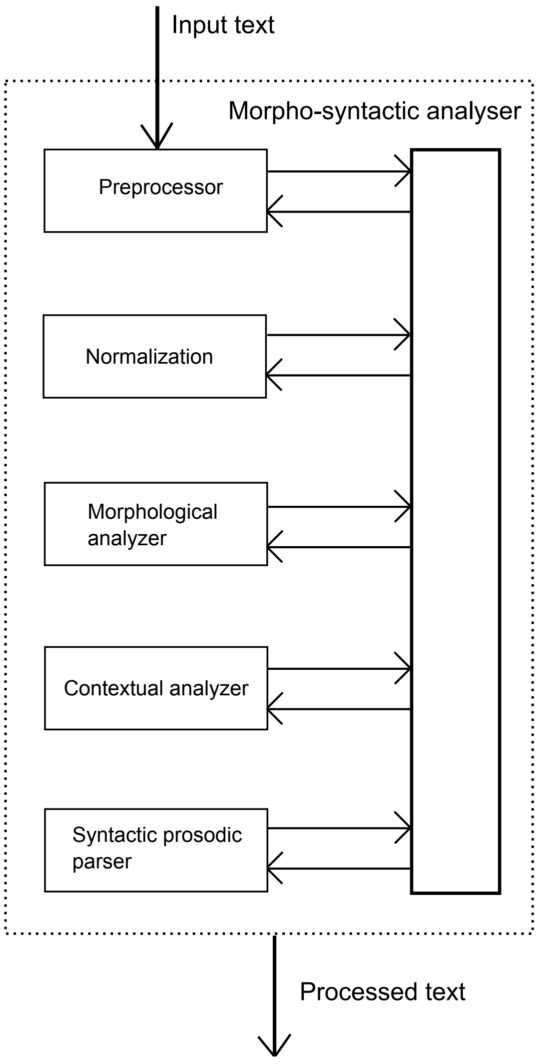


Figure 2.3: MSA module diagram[19]

detetcion of word stem and the type of word creation (using prepositions and endings) is required.

The contextual analysis

The contextual analysis module considers words in their context, which allows it to reduce the list of their possible part of speech categories obtained in the previous step to a very restricted number of highly probable hypotheses. There are two types of methods: stochatic methods and non-stochastic (deterministic) methods. Stochastic methods are based on transition probabilities between two neighbouring word categories. The probabilities can be computed explicitly (using n-grams) or implicitly (using neural networks). Deterministic methods are based on classification and regression tree (CART) techniques. They are using yes/no rules for accepting/rejecting certain combinations of syntactic categories.

The syntactic-prosodic parser

The syntactic-prosodic parser finds the text structure (i.e. its organization into clause and phrase-like constituents) which more closely relates to its expected prosodic realization. For example it can determine phrases on basis of positions of commas, dashes, etc.

2.1.2 Phonetic transcription

Phonetic transcription module converts the orthographical symbols into phonological ones using a phonetic alphabet (GTP conversion).

The GTP conversion can use two approaches: a dictionary-based approach and a rule-based approach:

- **The dictionary-based** approach is based on storing a phonetic dictionary which contains the grapheme representation of a word together with its phonetic transcritpion. This approach is very quick and accurate and the pronounciation quality is good but the major drawback is that it needs a large database to store all words and the system will stop if a word is not found in the dictionary. This approach is more suitable for analytic languages (e.g. English) in which there are not many forms derived from the same word (e.g. using inflection endings). Thanks to that the dictionary can be reasonably small.

- **The rule-based** approach is based on transcription rules. The main advantage is that it requires no database and it works on any type of input. This approach is more suitable for flexional languages (e.g. Czech) with many forms derived from the same word where the phonetical dictionary would become very big. It is necessary to find general rules for automatic transcription.These rules can be created analytically (often with help of human-expert) or automatically using machine learning on a representative training set.

Both approaches are often combined. For example for Czech, for native words rules are used and foreign words are stored in an exceptions dictionary.

The phone is a sound that has definite shape as a sound wave. Phone is the smallest sound unit. The set of phones that constitute minimal distinctive phonetic units is called phoneme. Phonemes have certain articulation and acoustic characteristics. These characteristics are then used for the speech synthesis.

Czech phonetics and phonology

This project is mainly focus on Czech. In Czech[30] there are two groups of phonemes: vowels and consonants. There are 10 vowels, 27 consonants and 3 diphtongs in Czech.

Vowels carry tone quality and estetic feeling of the speech. Acoustic signal of vowels have quasi-periodic course (with period of the fundamental frequency of the vocal chord - F_0) and higher amplitude. Vowels are different from each other in articulation and lenght. The articulation of vowels is determined by the vertical (low to hight) and the horizontal position (back to front) of the tongue and the lip position (roundedness). All vowels are voiced. The Czech classification of vowels is shown in table 2.1.

Diphtongs are a special case and are composed of two vowels (for Czech: ou, au and eu) and during the articulation of the utterance it smoothly transforms from the first part of the diphtong to the second part. However the second part is considerably supressed.

Vertical tongue position	Horisontal tongue position		
	Front	Central	Back
Close (high)	i, í		u, ú
Mid	e, é		o, ó
Open (low)		a, á	

Table 2.1: Vowels in Czech.

Consonants are the main part of the intelligibility of the speech. Acoustic signal of consonants have presence of a characteristic noise and compared to vowels the amplitude is also lower. Articulation of consonants is in virtue of place and manner of the articulation and the vocal chord activity (phonation). Consonants classification is showin in the table 2.2.

Manner	Place of articulation											
	Labial		Alveolar		Post-alveolar		Palatal		Velar		Glottal	
Nasal		m		n				ň				
Stop	p	b	t	d			ť	ď	k	g		
Affricative			c		č							
Fricative	f	v	s	z	š	ž			x			h
Trill			r, ř									
Approximant			l					j				
Voiced	-	+	-	+	-	+	-	+	-	+	-	+

Table 2.2: Consonants in Czech.

2.1.3 Prosody generation

The prosody generation is a very important part of the TTS system because the intelligibility and the naturalness of the synthesized speech is considerably dependend on the prosody characteristics contained in the speech. Also the focus is dependend on sentence prosody. For example, there are certain pitch events which make a syllable stand out within the utterance, and indirectly the word or syntactic group it belongs to will be highlighted as an important or new component in the meaning of that utterance. The presence of a focus marking may have various effects, such as semantical contrast, depending on the place where it occurs, or change the context of the utterance. The next two examples show how focus changes semantics of the same sentence:

I saw **him** yesterday.
I saw him **yesterday**.

In the first sentence the stress is on *him* and somebody cpuld say it, who is pointing on a particular person of the group. In the second sentence, the stress is on *yesterday* and it emphasizes the time of the event.

Prosody generation is a very complicated task because the prosody is partially independent on the text representation of the speech. Methods of prosody generation uses knowledge of the sentence parsing into syntactic-prosodic phrases given by the MSA module. The most important prosody characteristics are intensity, duration and intonation.

Intensity generation

Intensity generation is considered to be the least important part of prosody generation. The basic units of intensity generation are phonemes. Their intensity is determined using statistics based on large speech corpora in connection with specific information about the position in word, sentence, sonority, etc.

Duration generation

The duration generation in TTS systems includes the estimation of all aspects connected with duration like the duration of speech segments, pause positioning, rhytmic segmentation of utterances using accent, etc. Duration modulation basic units are phoneme level segments. Duration models of these segments use knowledge of the articulation and phonologic aspects of these segments in form of rules or statistics calculated on large speech corpora. Pause generation uses information about syntactic-prosodic sentence structure, because pauses are positioned on boundaries between the certain phrases. Pauses can have variable length depending on the context. For accent generation all three basic prosody characteristics - fundamental frequency, duration and intensity - need to be modified. In TTS systems usually only the word accent is modelled.

Intonation generation

Intonation generation is related to contour of fundamental frequency (F_0), and voice pitch in consequence. Intonation generation can be divided into two steps: the creation of a symbolic intonation description (complete prosody) of the speech as a complement to orthographic and phonetic transcription, and generation of resulting intonation contour (F_0 contour) using selected intonation model on basis of the created description. Used symbolic description generally determines which intonation models are appropriate for intonation generation.

All symbolic prosody descriptions are characteristic of effort of the best description of the prosody contour (especially intonation). The most frequent is limitation of description just for significant events in the F_0 contour like accents and ending tones. Description is exclusively made in acoustic domain. The most familiar prosody symbolic descriptions are ToBI, INTSINT and Tilt. The ToBI (tones and break indices) transcription system aside from accents and ending tones also marks boundaries between utterance segments. For marking ToBI uses set of symbols describing decline (L) and growth (H) in contour of F_0 contour. It distinguishes several symbols marking F_0 contour on stressed syllables ($H^*, L^*, L^* + H, L^*+!H, H+!H^*$ and $!H^*$). Further it marks declining ($L\%$) or growing ($H\%$) intonation contour of finished and unfinished (*L-, H-*) prosody phrases. Resulting intonation contour can be composed of one or more unfinished phrases and one finished phrase, for example *L-L%, L-H%, H-H%, H-L%* and so on. An example of a F_0 contour is shown in figure 2.4.

Resulting F_0 contour generation is done by intonation models. Intonation models can be categorized according to domain in which work with intonation is done:

- **Acoustic intonation models** – are based on acoustic representation of intonation via F_0 contour in time. This category is most widely used because of relatively easy record, measure and detection of F_0 contour. The most famous models are Fujisaki model and Tilt intonation model.

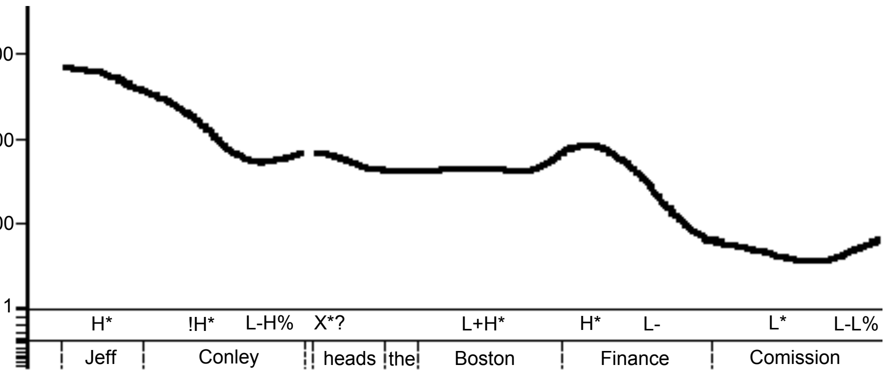


Figure 2.4: Example of F0 contour[7].

- **Perception intonation models** – these models are working with perception representation of intonation, i.e. on listener intonation perception level. This category is represented by perception stylisation models.

- **Linguistic intonation models** – these models are the most complicated intonation models because they are coming out from general linguistic prosody representation. The most famous representatives are pitch contour theory and tone sequence theory.

From the solely F_0 contour generation perspective intonation models can be categorized to:

- **Rule-based generation** – The simpliest generation models. A set of linguistically motivated rules for generating F_0 contour are designed based on selected intonation transcription description. Arguments of these rules can be e.g. synthetized speech pitch range, utterance type (declarative, tentative, etc.) or duration of individual phonemes. Parts of F_0 contour generation are words accent detection, estimate F_0 contour characteristics, sentence accent (duration) estimation, etc. Based on these data F_0 contour of the synthesized speech is generated. Because application of rules can cause steeper changes on individual intonation events boundaries, F_0 contour interpolation is performed. These models often use ToBI transcription system.

- **Parametric intonation models** – Parametric models work with similar information as rule-based models. The difference is that F_0 contour contour and its important events are represented parametricaly using properly selected parametric model. To this category a Fujisaki model or a Tilt model belong.

- **Corpus oriented generation** – For F_0 contour generation natural speech utterance corpora can be used. Based on utterances in the corpus F_0 generation parameters are automatically set. Intonation model works with F_0 contour inventory then. For inventory creation classification and regression trees (CART), statistical methods (neural networks or Hidden Markov models) or directly transcripted utterance (e.g. using ToBI) can be used. The simpliest model is prosody transplantation which saves directly individual F_0 contours of every utterance for a given speaker.

This categorization is not disjunctive though, often models overlaps over more categories. For example parametric models can use special heuristics, i.e. variously designed rules.

2.2 Speech synthesis

The speech synthesis module performs speech synthesis on the signal level, based on information from the NLP module. The speech synthesis can be categorized according to several aspects.

According to human participation degree synthesis can be divided to a rule-based synthesis and a data-driven synthesis. In case of the **rule-based synthesis** synthesizer parameters are set manually on a basis of a set of manualy derived rules. Traditionally representative of this category is a formant synthesizer. In the **data-driven synthesis** synthesizer parameters are set automatically on a basis of real speech data. A concatenation synthesis belongs to this category.

According to domain of signal processing synthesis can be divided to **time-domain synthesis** and **frequency-domain synthesis**. Categorization based on the speech representation distinct between **parametric methods** and **non-parametric methods**. Time-domain methods belongs to the non-parametric methods because they are working directly with time samples of the speech signal. Frequency-domain methods belongs to the parametric methods because they represents the speech using various (frequency) parameters.

Categories according to the way of modeling used during the resulting speech creation are:

- **Articulatory synthesis**
- **Formant synthesis**
- **Concatenative synthesis**

Target of the articulatory synthesis is to create a model of the whole speech production system (exact human vocal tract imitation). In contrary the formant synthesis and the concatenative synthesis concentrates on the speech signal and creating a model of it.

2.2.1 Articulatory synthesis

The articulatory synthesis uses a physical model of the vocal tract including e.g. articulators and their positioning. Signal is created using mathematic simulation of breathe air stream and individual articulation models activity simulation. Articulation parameters are relative positions of the individual articulators like size and shape of labial slot or height and position of the tongue. Excitation parameters are e.g. lung pressure and glottis size.

The articulatory synthesis represents the most general way of the speech synthesis. However because of being very complex this method is not very popular. Also the quality level of the synthesized speech doesn't reach the quality level produced by the formant or the concatenative synthesis methods. Articulation synthesis scope is limited rather on producing isolated sounds, phonemes or syllables now.

2.2.2 Formant synthesis

The formant synthesis is based on the source-filter theory. According to this theory the speech production is modelled with two independent components: an **excitation source** and a **linear acoustic filter** representing a frequency response of the vocal tract. Excitation source generates kvaziperiodic sequence of phone pulses for voiced sounds and random noise for unvoiced sounds. Speech production model is shown in figure 2.5.

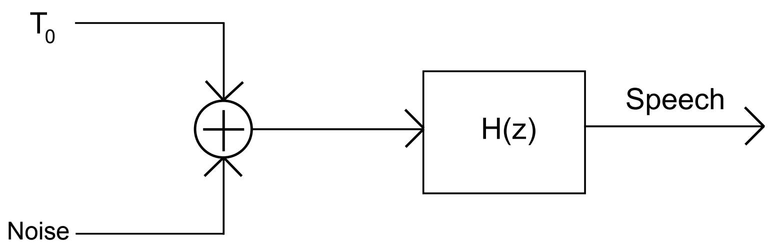


Figure 2.5: Simplified speech production model according to the source-filter theory[19].

Following the source-filter theory the formant synthesizer simulates simplified the human speech production:

- **Excitation source** – is composed of the voiced sounds source simulating F_0 frequency and of voiceless sounds generating noise. According to resulting sound characteristic voiced or voiceless source is used.
- **Vocal tract** – is simulated using filter which parameters are tied with human voice tract formants. Acoustic characteristics of the speech are modelled by the filter and consists of resonators.

The formant synthesis performs the synthesis using a set of rules. These rules transforms phonetic information on the system input (usually phonemes and prosody units) to the sequence of synthesizer parameters. These parameters are mainly formants and other acoustic or articulation parameters. Scheme of the formant synthesizer is shown in figure 2.6. The most famous representative is Klatt formant synthesizer.

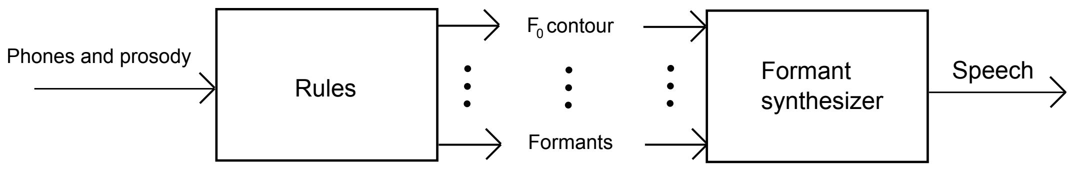


Figure 2.6: The rule-based formant synthesizer scheme[19].

Main advantages of the formant synthesis are small parameters number (minimal memory requirements), easy controlling prosody characteristics, constant quality of the synthesized speech and easy voice characteristics modification.

Main disadvantages are complicated manual parameters finding, low naturalness of the synthesized speech and mutual interactions between parameters.

2.2.3 Concatenative synthesis

The concatenative synthesis is the most widely used speech synthesis method nowadays. Basic principle of this method is based on assumption that individual sounds, from which the speech is generally composed, can be represented using finite number of speech (acoustic) units. The speech units are stored in a speech units inventory. This inventory is usually segmented speech corpus containing the speech units together with boundaries of individual speech units (the most frequently phones). The resulting synthesized speech is created using concatenation of the speech units. Scheme of the concatenative synthesis is shown in figure 2.7.

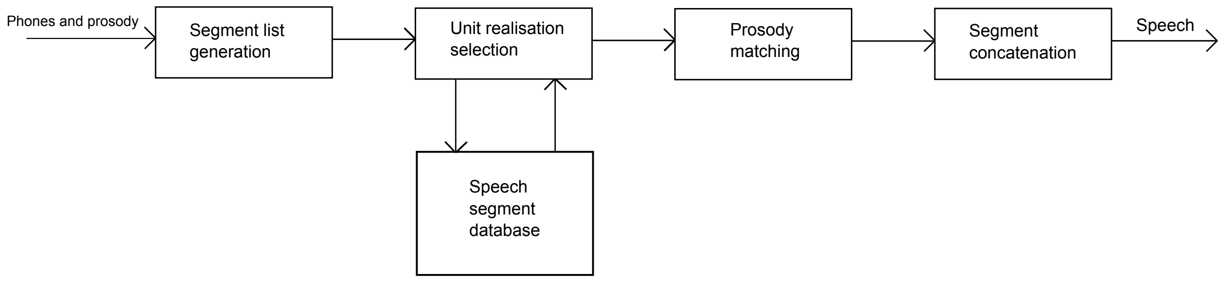


Figure 2.7: The concatenation synthesis scheme[19].

Main advantages are good synthesized speech quality, corpus speaker voice characteristics copying and quick synthesizer design.

Main disadvantages are computing and memory requirements, difficult voice characteristics modification and inconsitent quality of the synthesized speech.

Speech units

Speech units selection is the first task in of the design process of the concatenation synthesizer. Speech units should be selected such as to cover the whole sounds spectrum and can be easily concatenated at the same time.

There are many units that are used for the speech synthesis. **Words** are the least used units in TTS systems because very much words exist and their storage in computers is impossible now. Howewer word-based synthesised speech is very natural because of minimal connection points. **Syllables** are still relatively too long and their count in a language is substantial so they are practically not used. **Phonemes** represents the smallest units of the natural speech, their count is reasonable and arbitrary utterance can be producted using them. However there are many acoustic realisations of each phoneme dependend on the context so concatenation of phonemes is problematic. **Triphones** are units considering either left and right phoneme context. Triphones can comprise coarticulation features well which is important for the quality of the synthesized speech. Therefore triphones are used as the speech units mostly. Speech units illustration is shown in figure figUnits.

Words	sobota					
Syllables	so		bo		ta	
Phonemes	s	o	b	o	t	a
Triphones	<sil>-s+o	s - o + b	o - b + o	b - o + t	o - t + a	t - a + <sil>

Figure 2.8: The illustration of the segmentation of the word *sobota* on the speech units.

Speech units inventory

The speech units inventory represents list of all units and their acoustic realisations. For its creation it is required to have the speech corpus and to process a segmentation. The segmentation is the process of finding boundaries of the selected speech units in the speech data. The segmentation can be manual and automatic. The automatic segmentation is mostly used especially on large speech units inventory.

There are two mainly used methods of the automatic segmentation: a HMM-based method and a DTW-based method. Because of the better consistency of segmentation the HMM-based method are prefered.

Automatic segmentation with Hidden Markov models

The HMM-based segmentation is the mostly used method of the segmentation of the speech inventory due the good segmentation consistency. For segmentation mainly **multiple HMM** are used. For every model corresponds certain linguistic mark. Multiple HMM are used for selecteted speech units modelling then. The automatic segmentation scheme is shown in figure 2.9.

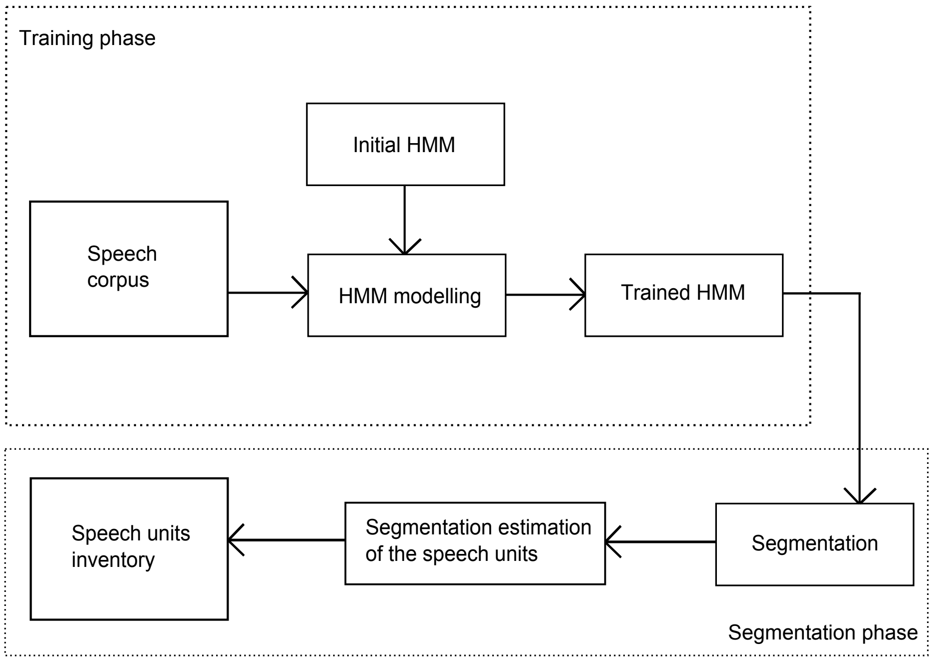


Figure 2.9: The automatic segmentation of the speech corpus using multiple HMM[19].

Before using HMM it is appropriate to do analysis of parameters for each utterance in the corpus and describe every utterance with sequence of vectors containing certain speech parameters (e.g. MFCC or PLP).

The segmentation is done in two phases:

- **Training phase** – Parameters estimation of individual models from the parametrized speech data is done during this phase. Parameters estimation of the models is subsequently improved during this iterative process. **Baum-Welch reestimation algorithm** is the most widely used for estimation. Starting estimation of the parameters can be set on the same value calculated as an average from all speech data in the corpus for all models. Results of this face are trained HMM of the speech units (models with estimated parameters).

- **Segmentation phase** – During this phase individual HMM are assigned to certain sequences of parameters vectors. This way segments modelled by individual HMM are identified in the speech data and boundaries between them are set. For this purpose **Viterbi algorithm** is the most frequently used.

Prosodical and spectral modifications methods

Because of the huge variability in phonetic and prosody characteristics of the human speech, no speech corpus can contain all speech contexts required during the speech producing. Therefore it is needed to modify the characteristics of the concatenated segments in a certain way often. There are two types of modifications:

- **Prosody modifications**
- **Spectral modifications**

Both modification types should approximate prosody and spectral characteristics of the selected units to the characteristics of the desired synthesized speech. Due to these modifications relatively small database can be used and still good quality of the synthesized speech can be achieved.

PSOLA

PSOLA (Pitch Synchronous Overlap and Add) belongs to the prosodical and spectral modifications methods and it is one of the most successful methods of speech synthesis. It was created as an extension of OLA (Overlap and Add), SOLA (Synchronous Overlap and Add) and mainly WSOLA (Waveform-Similarity based Overlap and Add) algorithms.

Methods based on PSOLA algorithm do not produce the speech directly. They allow to concatenate saved speech segments and change F_0 period and segments duration during it. They are used standalone in time domain (TD-PSOLA) or in connection with a parametric speech production method (MBROLA). All of them produces good quality resulting speech.

On input algorithm have the original (analytic) speech signal and a sequence of pitch marks. The pitch marks corresponds to the moments of the vocal tract excitation. Algorithm then works in three steps:

1. **Analysis** – Analysis performs speech signal decomposition to a sequence of short-term signals (STS) in moments defined by the pitch marks.
2. **Modification** – Modification transforms the sequence of analytic STS to the sequence of synthetic STS. Synthetic STS are synchronized with new time moments designed to satisfy required time and frequency characteristics.
3. **Synthesis** – The last step of the algorithm is the speech synthesis. The resulting speech signal is created combinating synthetic STS placed on new pitch marks positions.

Corpus-based synthesis

Corpus-based synthesis uses large speech corpora, often segmented on phone level and exhaustively phonetically described (with emphasis on spectral and prosody characteristics of individual phones). All operations during the speech synthesis the run automatically based on the selected speech corpus so the resulting speech quality directly depends on the corpus quality. Therefore provide quality enough speech corpus is necessary to achieve the good synthesis quality.

During the synthesis it is necessary to select the most appropriate (their phonetic, spectral and prosody context matches desired context of resulting speech mostly) speech units and concatenate them. Selecting such units guarantees the least discontinuities between neighboring units in the synthesized speech. During concatenation units can be used unmodified in the state as they are stored in the corpus or certain prosody and spectral modification can be done with them in order to improve the synthesized speech quality.

Unit selection

The unit selection is the most significant representative of the corpus-based synthesis without using descision trees. The unit selection method uses large speech corpus segmented to phonetically and prosodically phoneme type anotated units. Input information is a sequence of units for synthesis (usually phones or diphones) supplied with prosody characteristics (intonation, duration and intensity). This information describes desired speech output and is denoted as a target specification. The target of the algorithm is to find the most appropiate speech units in the speech corpus and to produce the resulting synthesized speech using the smoothened concatenation of them. The task lies in finding optimal sequence of speech units in the speech corpus within scope of synthesized utterance. It is assumed that the more precise units sequence algorithm finds, the less speech signal modification need to be done to produce the synthesized speech according to that specification. The resulting synthesized speech should sound more fluently and more naturally then.

During finding optimal units sequence two **cost functions** are considered. **Target cost** $C^t(t_i, u_i)$ describes difference between database unit u_i and and the target unit t_i which it should represent during synthesis. **Concatenation cost** $C^c(u_{i-1}, u_i)$ represents quality of neighboring units u_{i-1}, u_i concatenation. For successfull acomplishment of the target right definiton of both cost functions and balanced parameters estimation is required. Optimal sequence finding is shown in figure 2.10.

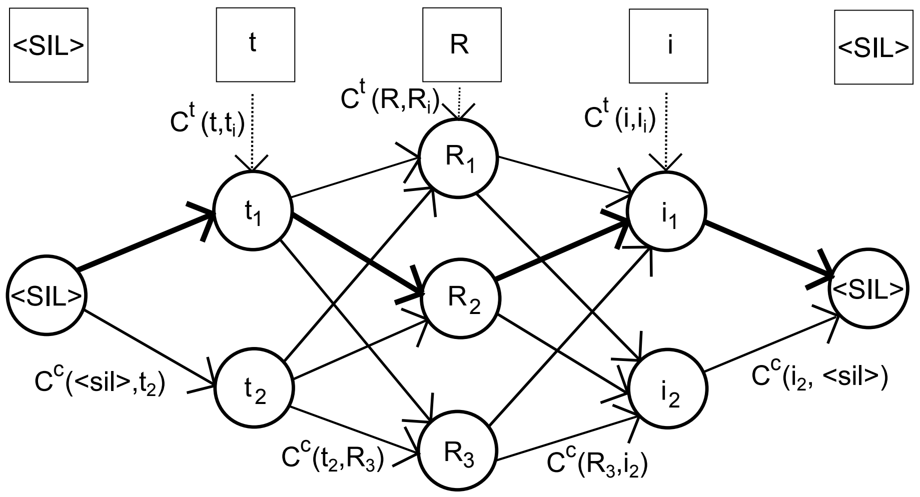


Figure 2.10: Optimal sequence finding illustration for sentence *tři*. Symbol <*sil*> marks a pause. Bold lines illustrates the optimal path.

Binary phonetic descision tree-based HMM clustering

Binary phonetic descision tree-based clustering was used for a speech recognition at first. Later it turned out that this algorithm is suitable for the automatic unit selection usable in the speech synthesis too. During segmentation process it is possible to define units directly intended for the speech synthesis using Hidden Markov models of phones (or triphones eventually). Binary phonetic descision tree-based HMM clustering is used to achieve that. Afterwards the segmentation doesn't necessarily need to be processed on general phones level but directly on units defined during clustering.

During the HMM segmentation every unit is segmented with a certain probability. The higher the probability the more similar the unit to the desired unit is (assuming representative speech corpus is used). This probability can be used as a simple unit selection criterion for the speech synthesis.

Binary phonetic descision tree-based HMM clustering algorithm can run on two levels:

- **The whole model level (i.e. phoneme level)**
- **The model state level (i.e. sub-phoneme level)**

For the speech synthesis the sub-phoneme level is not suitable because sub-phoneme units are too short and their concatenation cause a large amount of junction points (potentionally concatenation synthesis incontinuity sources). Therefore clustering is better run on the phoneme level.

Phonetic descision tree is a binary tree which descision (non-leaf) nodes have questions assigned. The question query left or right phonetic context and demands answer of yes/no type. Examples of such questions can be: R-silence? („is the previous phoneme a silence?“) or L-vowel? („is the next phoneme vowels?“). Simple questions querying only the nearest neighbour are used. Query can involve arbitrary phonetic context. Phonetic descision tree is illustrated in figure 2.11.

Before algorithm start trained individual contextually dependend models (i.e. triphone models) have to be available. Binary phonetic descision tree-based clustering algorithm can be written as follows:

1. For every phone (eventually phoneme) i exists N contextully dependent models (triphones) $M_k^i, k = 1, \ldots, N_i$. At start models M_k^i are placed to the root of the tree. For one phone also one binary descision tree will exist. The root is marked as a dividing node of the tree.
2. Consecutively apply all questions from the prepared questions list on the dividing node. Every question divides this node to two subnodes. In doing so the size of division gain is calculated as a difference of selected cost function before and after division.
3. The question with the biggest gain is selected and assigned to the dividing node. All models from from the parent node are divided to two groups of sub-nodes by this question. In doing so it is checked if at least one of terminating criterions is not fullfied:
   (a) **Sufficient probability gain criterion** – if the gain of the selected question is lower than chosen threshold value, the dividing node is declared as the leaf node. This criterion limits „useless“ tree division.

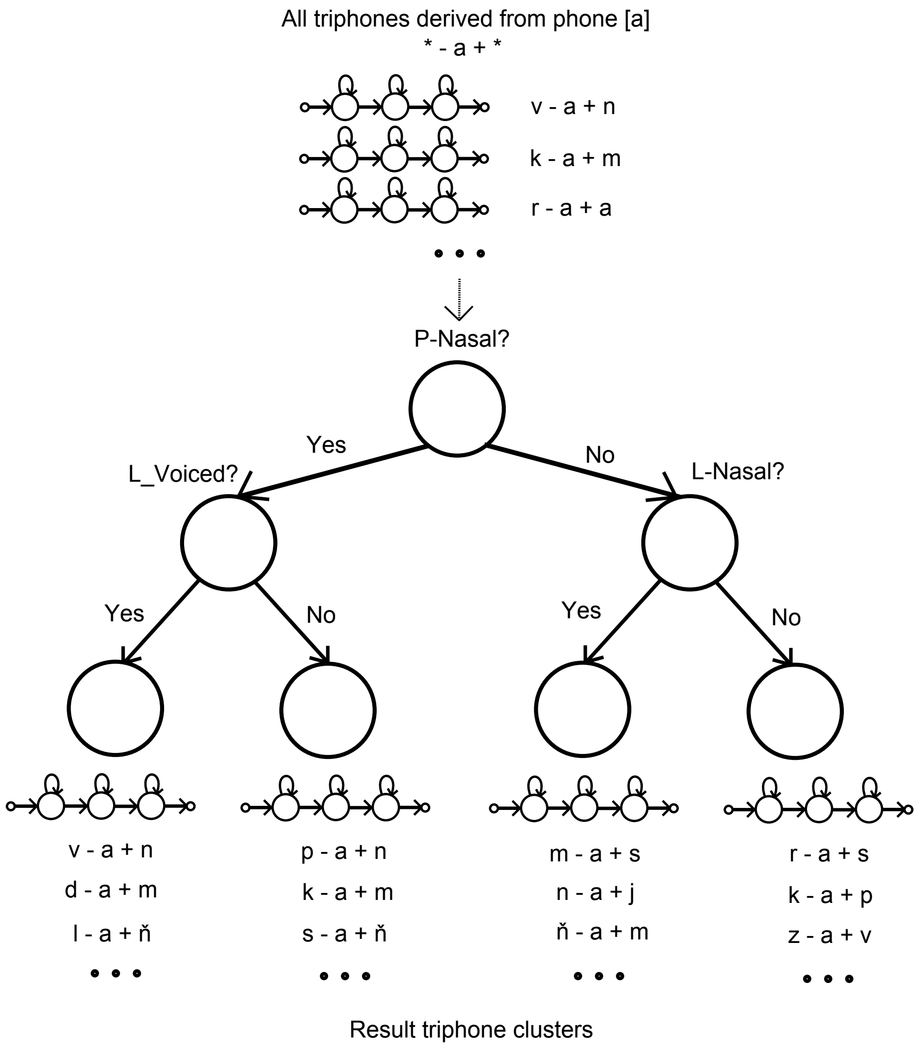


Figure 2.11: Illustration of finding the Optimal sequence for utterance *tři*. The symbol <*sil*> marks a pause. Bold lines illustrates the optimal path[19].

(b) **Sufficient data amount criterion** – if the count of training vectors in at least one of the resulting sub-nodes drops under chosen threshold value the dividing node is declared as the leaf node. This way sufficient amount of data in the newly created cluster is guaranteed.

4. Select a non-leaf node which do not have any question assigned yet and mark it as the dividing node. If there is no such node continue on 6.

5. Continue on 2., repeat until at least one criterion in 3. is not satisfied.

6. Optional result clusters (leafs of the phonetic descision tree) clustering. Cluster all couple of clusters into one if their clustering brings lower probability declination than the sufficient probability gain criterion threshold.

Results of of the algorithm are P_i clusters $S^i_p, p = 1, \ldots, P_i$ for all triphone models M^i_k derived from the same basic contextually independent phone i model. These clusters correspond to individual binary descision tree nodes.

Chapter 3

The MARY TTS system

This chapter describes the MARY (Modular Architecture for Research on speech sYnthesis) text-to-speech system [23] used in this project as a basis for further development. In the following sections the overall architecture of the MARY and the individual modules are described.

MARY is an open-source [8], multilingual text-to-speech synthesis platform written in Java. It is a flexible tool for research, development and teaching in the domain of text-to-speech (TTS) synthesis because it allows for a step-by-step processing with access to partial processing results. Thereby, the user is given the opportunity to interactively explore the effects of a specific piece of information on the output of a given processing step. MARY also comes with toolkits for adding new languages and for building unit selection and HMM-based synthesised voices.

3.1 Overall architecture

In principle, the modular design of the MARY system allows arbitrary system architectures. An architectural frame is defined via the notion of data types, specifying the data format serving as input and/or output to processing modules. Each module knows about the data type it requires as its input and the data type it produces as its output. Two modules can be „plugged“ together if the first module's output type is identical to the second module's input type.

Figure 3.1 shows the individual processing modules, the flow of information and the intermediate results corresponding to data types defining the interfaces between the modules. In the following, each of the modules will be briefly presented.

3.2 System modules

3.2.1 Plain text

Plain text is the most basic, and maybe most common input format. Nothing is known about the structure or meaning of the text. The text is embedded into a MaryXML document for the following processing steps.

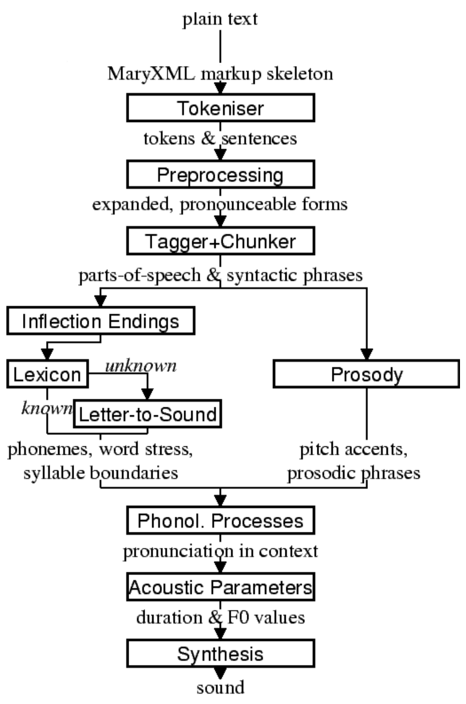


Figure 3.1: MARY TTS system architecture diagram

3.2.2 MaryXML markup

MaryXML is an internal, relatively low-level markup which reflects the modelling capabilities of this particular TtS system. MaryXML is based on XML. A MaryXML Schema formally specifies the structure of a correct MaryXML document. This first MaryXML skeleton, before tokenisation, is not required to comply to the MaryXML Schema which assumes text data to be tokenised. All subsequent intermediate MaryXML results (module outputs) do comply to the MaryXML Schema.

3.2.3 Tokeniser module

The tokeniser cuts the text into tokens, i.e. words and punctuation marks. It uses a set of rules determined through corpus analysis to label the meaning of dots based on the surrounding context.

3.2.4 Text normalisation module

In the preprocessing module, those tokens for which the spoken form does not entirely correspond to the written form are replaced by a more pronounceable form.

Numbers

The pronunciation of numbers highly depends on their meaning. Different number types, such as cardinal and ordinal numbers, currency amounts, or telephone numbers, must be identified as such, either from input markup or from context, and replaced by appropriate token strings.

Abbreviations

Two main groups of abbreviations are distinguished: Those that are spelled out, such as „USA“, and those that need expansion. The first group of appreviations are correctly pronounced by spelling rules. The second group is pronounced using an expansion table, containing a graphemic and optionally a phonemic expansion.

3.2.5 Part-of-speech tagger and chunker

Part-of-speech tagging is performed with the statistical tagger TnT, using the Stuttgart-Tübingen Tagset (STTS), and trained on the manually annotated NEGRA corpus. A chunk parser is used to determine the boundaries of noun phrases, prepositional phrases and adjective phrases.

3.2.6 Inflection endings

This module deals with the ordinals and abbreviations which have been marked during preprocessing as requiring an appropriate inflection ending. The part-of-speech information added by the tagger tells whether the token is an adverb or an adjective. In addition, information about the boundaries of noun phrases has been provided by the chunker, which is relevant for adjectives.

3.2.7 Lexicon

The pronunciation lexicon is language specific and contains the graphemic form, a phonemic transcription, a special marking for adjectives, and some inflection information.

3.2.8 Letter-to-sound conversion

Unknown words that cannot be phonemised with the help of the lexicon are analysed by a „letter-to-sound conversion“ algorithm.

3.2.9 Phonemisation output

The output of the phonemisation component contains the phonemic transcription for each token, as well as the source of this transcription (simple lexicon lookup, lexicon lookup with compound analysis, letter-to-sound rules, etc.).

3.2.10 Prosody module

Prosody is modelled using GToBI, an adaptation of ToBI („Tones and Break Indices“) for German. ToBI describes intonation in terms of fundamental frequency (F0) target points, distinguishing between accents associated with prominent words and boundary tones associated with the end of a phrase. The size of a phrase break is encoded in break indices. Within Mary, break indices are used as follows: „2“ is a potential boundary location (which might be „stepped up“ and thus realised by some phonological process later on); „3“ denotes an intermediate phrase break; „4“ is used for intra-sentential phrase breaks; „5“ and „6“ (not part of GToBI) represent sentence-final and paragraph-final boundaries.

After determining the location of prosodic boundaries and pitch accents, the actual tones are assigned according to sentence type (declarative, interrogative-W, interrogative-Yes-No and exclamative). For each sentence type, pitch accent tones, intermediate phrase boundary tones and intonation phrase boundary tones are assigned. The last accent and intonation phrase tone in a sentence is usually different from the rest, in order to account for sentence-final intonation patterns.

3.2.11 Postlexical phonological rules module

Once the words are transcribed in a standard phonemic string including syllable boundaries and lexical stress on the one hand, and the prosody labels for pitch accents and prosodic phrase boundaries are assigned on the other hand, the resulting phonological representation can be re-structured by a number of phonological rules. These rules operate on the basis of phonological context information such as pitch accent, word stress, the phrasal domain or, optionally, requested articulation precision. Segment-based rules can be applied, such as the elision of Schwa in the endings „-en“ and „-em“, the backward assimilation of articulation place for nasal consonants, and the insertion of glottal stops before vowels of pitch-accented syllables with a free onset. However, with diphone speech such reductions seem to limit the intelligibility, so that they are deactivated by default.

3.2.12 Linguistically maximally rich MaryXML structure

The output of the postlexical phonological rules module gives a rich MaryXML structure, containing all the information added to the structure by all of the preceding modules.

3.2.13 Calculation of acoustic parameters module

This module performs the translation from the symbolic to the physical domain. The MaryXML structure is interpreted by duration rules and GToBI realisation rules. The duration rules are a version of the Klatt rules adapted for German, by fitting the rule parameters to data from the Kiel Corpus.

The realisation of GToBI tones uses a set of target points for each tone symbol. These targets are positioned, on the time axis, relative to the nucleus of the syllable they are attached to; on the frequency axis, they are positioned relative to a descending pair of topline and baseline representing the highest and lowest possible frequency at a given moment. The fact that these lines are descending accounts for declination effects, i.e. overall F0 level is higher at the beginning of a phrase than close to the end. As an example, the GToBI accent „L+H*“, associated with the syllable ['fUn] of the sequence [g@-'fUn-d@n] („found“) is realised as a target on the baseline at the start of the Schwa of [g@], followed by a target on the topline in the middle of the [U] in ['fUn]. Obviously, the actual frequency values of the topline and baseline need to be set appropriately for the voice to be used during synthesis, in particular according to the sex of the speaker.

3.2.14 Phone segment and acoustic parameter list: MBROLA input

The output produced by the calculation of acoustic parameters module is a maximal MaryXML structure, which can be used e.g. to derive timing information for synchronizing speech with taking heads and embodied conversational agents.

The structure can also be reduced to more simple synthesizer input, e.g. a list containing the individual segments with their durations as well as F0 targets, a format compatible with the MBROLA .pho input files.

3.2.15 Synthesis module

Among others, MBROLA is used for synthesising the utterance based on the output of the preceding module. Several diphone sets for a number of male and female voices can be used.

MARY also contains basic unit selection code, based on the cluster unit selection code taken from FreeTTS.

3.2.16 Sound output

Several audio formats can be generated, including 16 bit wav, aiff, au, and mp3.

3.3 Technical details

For usage system is designed as a client-server application. The system is composed of a main server or „manager“ program, a number of modules doing the actual processing, and a client sending input data and receiving processing results. System is multi-platform due to usage of Java runtime environment.

3.4 MaryXML

The MARY system uses an internal XML-based representation language called MaryXML. The purpose of an XML-based representation language is to serve as the data representation format inside a TTS system. For that reason, the concepts represented in it are low-level, detailed, and specific to the design decisions, modules, and scientific theories underlying the TTS system. By means of the Document Object Model (DOM), a standardised object-oriented representation of an XML document, the TTS system modules can operate directly on the XML document, interpreting and adding information.

The MaryXML syntax was designed to maintain a certain degree of readability for the human user, by keeping information redundancy at a minimum. Example of MaryXML file is showed in Listing 3.1.

Listing 3.1: Example of MaryXML showing output of phonemisation module for sentence „Vítejte ve světě řečové syntézy!“

```
<?xml version="1.0" encoding="UTF-8"?>
<maryxml xmlns="http://mary.dfki.de/2002/MaryXML" xml:lang="cs">
  <p>
    <s>
      <phrase>
        <t g2p_method="rules" ph="' v i: - t e j - t e" pos="content">
          Vítejte
        </t>
        <t g2p_method="lexicon" ph="v e" pos="function">
          ve
        </t>
        <t g2p_method="lexicon" ph="s v j e - T e" pos="content">
          světě
        </t>
        <t g2p_method="rules" ph="' R e - C o - v e:" pos="content">
          řečové
        </t>
        <t g2p_method="rules" ph="' s i n - t e: - z i" pos="content">
          syntézy
        </t>
        <t pos="$PUNCT">
          !
        </t>
      </phrase>
    </s>
  </p>
</maryxml>
```

Chapter 4

Creation of a new language and a new voice for the MARY TTS system

This chapter describes the data preparation to train the HMM models for the new voice and the new language module and the new voice was added to the MARY system.

4.1 Speech database

In order to train HMM models for speech synthesis, a database has to be created and the speech database recorded. For each speech utterance its text anotation need to be present to transcribe the speech sample and to train the HMM models properly.

4.1.1 Creating the database

For the database creation, certain text data have to be selected[20]. There are several factors that need to be considered when creating the corpus, like the purpose of the corpus, size of the corpus, the number and diversity of speakers, the environment and the style of recording, etc. Another requirement for the corpus is the phonetic coverage. The corpus can have a phonetic coverage close to natural speech, or phonetically rich coverage with uniform relative frequency of phonetic units (e.g. phonemes or diphones). Phonemes should be ideally present in possible all contexts (e.g. all diphone combinations). Also, the selected sentences should be reasonably short and easy to read.

For the purpose of this project, a small database with only one speaker was created. The data for the corpus were prepared using the MARY system's building tools. A more detailed description of collecting the data for the corpus is given in section 4.2.

The created database corpus consists of 1000 sentences composed of 45 925 phonemes. The phoneme statistic was counted using the created MARY GTP module for phoneme transcription of the selected text sentences. The phoneme distribution is shown in table 4.1. It shows that the corpus is not phonetically rich with the balanced phoneme coverage but rather follows the distribution of general Czech. The frequency of phonemes in the corpus corresponds relatively to the frequency of graphemes in Czech. The grapheme distribution in Czech[1] is shown in table 4.2. Because there is a strong relation between spelling and pronunciation in Czech, grapheme distribution can be also used for informative purposes.

Phoneme	Frequency	Coverage [%]	Phoneme	Frequency	Coverage [%]
e	4626	10,07	e:	596	1,3
o	3220	7,01	S	585	1,27
a	2945	6,41	h	548	1,19
i	2774	6,04	R	536	1,17
t	2566	5,59	f	513	1,12
s	2261	4,92	C	489	1,06
n	2005	4,37	x	480	1,05
l	1954	4,25	Z	419	0,91
i:	1776	3,87	T	403	0,88
k	1624	3,54	ou	356	0,78
v	1598	3,48	u:	303	0,66
r	1567	3,41	D	250	0,54
j	1496	3,26	g	232	0,51
m	1403	3,05	au	76	0,17
p	1397	3,04	o:	70	0,15
d	1319	2,87	eu	50	0,11
u	1118	2,43	dZ	34	0,07
N	1053	2,29	tS	12	0,03
a:	978	2,13	ts	5	0,01
z	856	1,86	@	2	0
b	817	1,78	dz	2	0
c	611	1,33			

Table 4.1: Phoneme distribution in the database.

Grapheme	Coverage [%]	Grapheme	Coverage [%]	Grapheme	Coverage [%]
o	8.66	m	3.22	ž	0.99
e	7.69	u	3.14	č	0.94
n	6.53	á	2.23	š	0.80
a	6.21	z	2.19	ů	0.69
t	5.72	j	2.11	f	0.27
v	4.66	y	1.90	g	0.27
s	4.51	ě	1.64	ú	0.10
i	4.35	c	1.60	ň	0.08
l	3.84	b	1.55	x	0.07
k	3.73	é	1.33	ť	0.04
r	3.69	h	1.27	ó	0.03
d	3.60	ř	1.21	ď	0.02
p	3.41	ch	1.17	w	0.01
í	3.26	ý	1.07	q	0.00

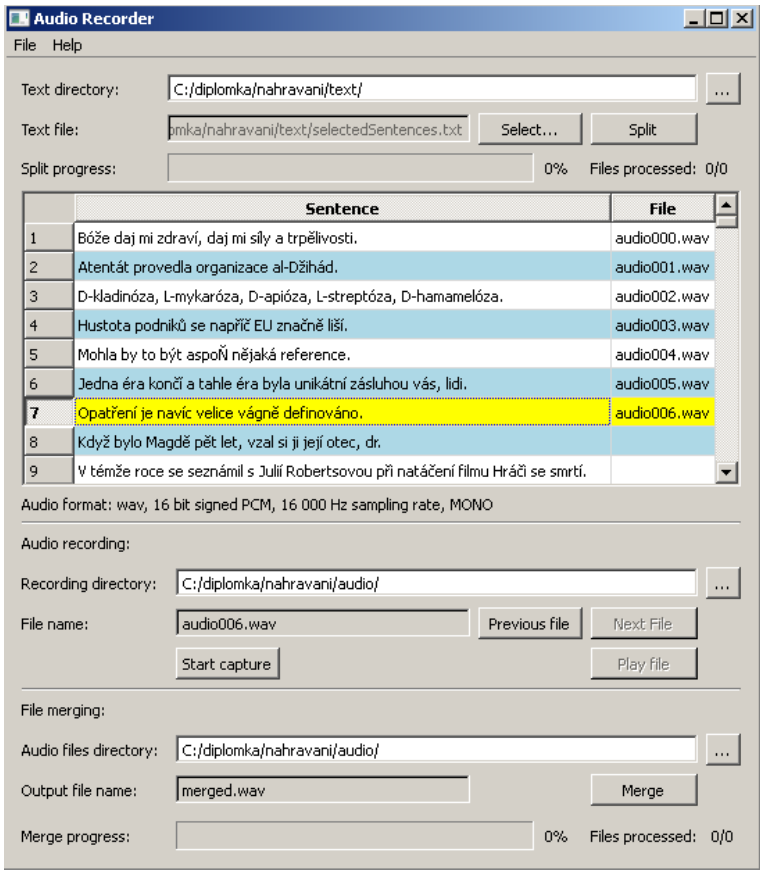
Table 4.2: Grapheme distribution in Czech. These statistics were calculated on text extent 3 139 926 graphemes[1].

4.1.2 Recording the speech database

Using the created text database the corresponding speech database was recorded. It is rather small (1000 utterances, about 1 hour of speech, 1 speaker) but sufficient for training HMM models and to synthesise intelligible speech. We chose to record the speech as isolated utterances. Each utterance is saved into a wav file corresponding to the text file including the text form of the utterance.

We recorded using a headset with noise cancelation (Plantronics Blackwire C620-M) to supress sound from the surrounding environment. In order to prevent corrupted utterances caused by fatigue or bad concentration a maximum of about 200 utterances was recorded per half-day. The format of the audio data is the following: 16 kHz sampling rate, 16-bit samples, mono wav files.

To create the speech database, an audio recorder program was implemented. It was implemented in the C++ programming language using the Qt toolkit[15] for GUI creation. For audio capture it uses the OpenAL Soft library[14]. It allows to load a text file with selected sentences for recording into the application and to show or to split the text file into new text files, where each file contains one sentence. The audio recording is done into the selected file, which then could be played to check the quality of the recording and to repeat the sentence if needed. The program also enables to merge all audio files into one file. A screenshot of the program is shown in figure 4.1.



Text directory: C:/diplomka/nahravani/text/

Text file: omka/nahravani/text/selectedSentences.txt

Select...

Split

Split progress: 0% Files processed: 0/0

	Sentence	File
1	Bóže daj mi zdraví, daj mi síly a trpělivosti.	audio000.wav
2	Atentát provedla organizace al-Džihád.	audio001.wav
3	D-kladinóza, L-mykaróza, D-apióza, L-streptóza, D-hamamelóza.	audio002.wav
4	Hustota podniků se napříč EU značně liší.	audio003.wav
5	Mohla by to být aspoŇ nějaká reference.	audio004.wav
6	Jedna éra končí a tahle éra byla unikátní zásluhou vás, lidi.	audio005.wav
7	Opatření je navíc velice vágně definováno.	audio006.wav
8	Když bylo Magdě pět let, vzal si ji její otec, dr.	
9	V témže roce se seznámil s Julií Robertsovou při natáčení filmu Hráči se smrtí.	

Audio format: wav, 16 bit signed PCM, 16 000 Hz sampling rate, MONO

Audio recording:

Recording directory: C:/diplomka/nahravani/audio/

File name: audio006.wav

Previous file

Next File

Start capture

Play file

File merging:

Audio files directory: C:/diplomka/nahravani/audio/

Output file name: merged.wav

Merge

Merge progress: 0% Files processed: 0/0

Figure 4.1: Audio recorder application

4.2 Integrating a new language into the MARY system

To add the a language into the MARY system the guideline[13] was followed. In the following subsections, the whole process of adding the new language is described.

4.2.1 Getting the MARY system running

Before starting the work some requirements have to be met:

- Linux like system or MSYS with MinGW[10] installed on Windows system
- MySQL[11] installed
- GIT tool (e.g. GITExtensions[2]) installed in order to get the source code of the MARY system and release completed work
- Java runtime environment installed
- Development environment for Java (e.g. NetBeans[12]) with Maven[9] support installed.
- All used text files need to be in UTF-8 encoding, otherwise errors can occur.

The first step is to get the MARY system running with the help of the above mentioned programs. First we had to get the source code from the GIT and then we had to build the project using the properly set development environment. We get the binaries of the Marytts-builder. These tools are used for the further work. Also the MARY server and the client applications were built and can be used to the functionality of the newly added components.

4.2.2 Creating the new language module

To add support for the new language, a minimal natural language processing (NLP) component needs to be created. The MARY Transcription tool was used to create this component. The tool provides a semi-automatic procedure for transcribing the text corpus in the new language and automatic training of Grapheme-to-phoneme (GTP) rules for that language. For this tool three things are required: a complete set of phones with the articulation features, dictionary containing words with their phonetic transcriptions and list of functional words for the target language.

- Phone set was created according to the Czech phonology described in chapter 2 using the SAMPA phonetic alphabet [16]. The phone set was written in the XML format required by the MARY system. The XML file is shown in listing B.1.
- The list of functional words was created according to the Czech grammar. Functional words are words that have little lexical meaning or have ambiguous meaning, but instead serve to express grammatical relationships of other words within a sentence. To the class of functional words belong e.g. prepositions, conjuctions and articles. In this project these words were used to build a simple part-of-speech (POS) tagger. A POS tagger classifies parts of sentences into categories and tags them. The POS tagger influences the prosody generation and the addition of POS tags improves the naturalness of the synthesized speech [22]. In this project a very simple POS tagger was created with only two categories: functional and content words.

- There are two ways of creating a pronunciation:
  - manually create the phonetic transcriptions for a provided list of words and incrementally train the GTP model and correct the results until the dictionary is sufficiently precise and coverage big enough (MARY guideline suggests using the most frequent words extracted from a Wikipedia dump, as described in the next subsection)
  - provide a complete dictionary from which a binary program for the NLP component module can be build immediately (assuming dictionary contains enough data to get satisfying result).

  In this project the second approach was used using dictionary created at FIT BUT[1]. Since the dictionary could not be applied out of the box, some changes were necessary. First of all, words with more than one transcription were eliminated to have just one transcription in order to prevent nondeterminism (and compilation errors when building NLP component in consequence). The transcription to be retained was selected manually according to other words with similar pronunciation. The next step is a syllabification of the phonemes. The syllabification is important for generating the prosody characteristics and has great influence on the quality of the resulting speech. The syllabification process is more described in chapter 5. The next step was tagging functional words in the dictionary using the created list. The last step was to transform the format of the dictionary in the following way:

$$ALE \quad a\ l,e \rightarrow Ale\ a\text{-}le\ functional.$$

  where symbols *,* and - marks the syllable boundary.

Several programs and scripts were written for these editations and transformations.

Using the inputs generated in the format described above, the binary NLP component was built using the Transcription tool. With the output of this procedure, a new language module for the MARY system was created. Following the directory structure according to other existing language modules, the NLP components and the dictionary was added and all paths were accordingly set. After successfully compiling the new module into the MARY system, the phoneme output from the module needs to be tested. If the phoneme output is correct, and no system errors occured, the new language module can be used as a part of the MARY system for usage, as well as a basis for the next steps. Also the newly created language module can be published for the future MARY TTS official releases.

4.2.3 Creating the database corpus from a Wikipedia dump

To create the database corpus, the MARY builder tools were used. As a text corpus serving as basis for the further steps the whole Wikipedia dump of articles in the Czech language was downloaded. This dump contains enough data for database selection and also contains both ordinary and specific text sentences.

Since the Wikipedia dump is in XML format, it is necessary to extract clean text from it. Since this point a MySQL database server is needed to be running and has to be properly set up. A database is created and the clean text is inserted into it to calculate the words

[1]Created by Ing. Kamil Chalupníček (http://www.fit.vutbr.cz/ chalupni/).

frequency. From this data in the database, only the most frequent words are transcribed and can be used for the manual creation of the minimal NLP components if there is no dictionary with phonetic transcriptions available. Given the minimal NLP components, feature marking procedure can be run. The MARY components for the new language need to be accesible by the MARY server. At this step sentences with unknown words or symbols are discarded and from the rest context features are extracted. These features are then used by a database selector, which selects a chosen number of a phonetically balanced sentences for the database corpus and saves it into the database. This corpus then can be saved into a text file using the SQL query:

```
SELECT CONVERT(cs_test_selectedsentences.sentence USING utf8)
FROM cs_test_selectedsentences
INTO OUTFILE '~/selectedSentences.txt'
```

Due to the large amount of data processed, this step takes a long time and some errors in the MARY system can occur during it. Sometimes editing the source code of the MARY system, with the help of the MARY development forum, was necessary. As a result of this step, the database corpus for the speech database recording was created.

4.3 The integration of a new voice into the MARY system

To add the voice into the MARY system, the guideline for creating new HMM voice[3] was followed. Adding new voice can be done on a Windows system with MSYS installed, but a linux like system is strongly recommended for the procedure because of many programs and many paths which are required to be set up properly.

A script for checking all necessary programs by MARY and the MARY Voice import tools were used to create the new voice. The script provides automated checking if all necessary programs are set up properly and if not, it downloads them and install them automatically. The Voice import tools provide automated step by step creation of the new voice.

- First of all data from the created database need to be prepared in the following pattern: one sentence per file, speech sound in wav file, text representation of the sentence in txt file and the names of wav and txt files have to mathch due to the transcription alignment calculations, e.g. files audio000.wav and audio000.txt need to contain the same sentence. Sound files have to be in directory ./wav, text files in directory ./text placed in selected working directory which will be used further for the procedure.

- Then the checking script have to be run in order to setup environment properly. Before running the Voice import tools the MARY server with the created new language module have to be started and running during certains steps. The Voice import tools together with installed programs then prepare the data for the HTS toolkit and HMM models are trained according to the characteristics of the created database. This procedure takes a long time and the output models and trees are used for the new voice creation.

For training HMM models from the speech database HMM tolkits were used. The Hidden Markov Model Toolkit (HTK)[4] is a portable toolkit for building and manipulating hidden Markov models. HTK is primarily used for speech recognition research although it has been used for numerous other applications including research into speech synthesis, character recognition and DNA sequencing. HTK consists of a set of library modules and tools available in C source form. The tools provide sophisticated facilities for speech analysis, HMM training, testing and results analysis. The HMM-based Speech Synthesis System (HTS)[5] has been developed by the HTS working group. The training part of HTS has been implemented as a modified version of HTK and released as a form of patch code to HTK.

With the created speech database and the Voice importing tools together with the HTS toolkit a new voice was created. After this process the new voice can be compiled into a jar file which can be together with the created new language module used in the MARY server for text-to-speech synthesis.

4.4 MARY client application

For demonstration of the result funcionality two client programs were created. The first program is a console application written in C++ using the OpenAL Soft library for audio playback and the Libcurl library[6] for HTTP communication with the MARY server. The MARY server is required to be already running on system before starting the program. It expects a text sentence as input in the command line, it creates a request for the MARY server, saves the returned resulting speech as the wav file and plays it to the user.

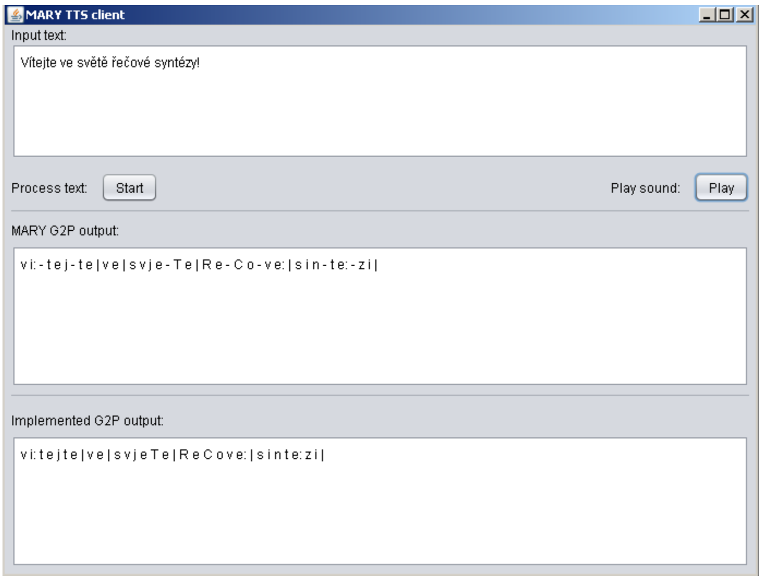
The second program is a GUI application written in Java. The reasons for switching to Java were easier integration with the MARY system (the MARY server is started embedded together with the program) and better system independency of the program (C++ together with the Qt toolkit and libraries for the HTTP communication and the audio playback turns out quite problematic). Also in Java problems with UTF-8 coding of diacritics for the HTTP request for the MARY server were solved.

Program expects a text input in the top text area. Button Start sends request to the MARY server for a phoneme output and shows it in the middle text area. Button Play sends request to the MARY server for an audio output and plays it as a speech sound. As a part of the program the GTP module is implemented. Its output is shown in the bottom text area. A screenshot of the program is shown in the figure 4.2.

4.5 Evaluation of the synthesized speech quality

In order to evaluate the resulting quality of the TTS systems there is a need to deal with the synthesized speech of arbitrary content, so comparing synthesized with original speech in order to evaluate the quality of the synthesis is an almost impossible goal[19]. Because there are no objective tests, listening tests come in place. The most common way is to test on a group of listeners who subjectively evaluate the synthesized speech[26].

During evaluation the intelligibility and the naturalness of the synthesized speech are tested. The intelligibility tests are evaluating how listeners understands the synthesized speech with an emphasis on percepting transition sounds – coarticulation. These tests are limited to the segmental speech quality. The naturalness test are evaluating the speech from an overall aspect. These test are testing also the suprasegmental quality.



Vítejte ve světě řečové syntézy!

```
vi:-tej-te|ve|svje-Te|Re-Co-ve:|sin-te:-zi|
```

```
vi:tejte|ve|svjeTe|ReCove:|sinte:zi|
```

Figure 4.2: Mary client for demonstration of functionality.

4.5.1 Modified rhyme test

Modified rhyme test (MRT) was used to test the intelligibility of the synthesized speech. MRT is one of the most widely used intelligibility tests. During the test one word is played to the listener and his task is to identify it in a group of 6 similar words. The test is concentrated on consonants because the synthesis of consonants has the biggest influence on the intelligibility of the speech.

Words in groups are always one-syllable and differ in the first or the last consonant. After evaluating enough number of words a result diagnostics can be made. For the test in the project 40 groups of words were selected[25],[27]. They are shown in the table 4.3.

Advantages of the MRT are reliability, relatively small number of listeners needed to perform the test and possibility to compare results with the other synthesizers. Listeners can also be „unqualified“. Disadvantages are the test limitation on the first and the last consonants and the limited number of alternatives in each group so listeners can make their descisions according to the words list.

4.5.2 Mean opinion score test

Mean opinion score (MOS) test was used to test the naturalness of the synthesized speech. The test results from the subjective rating of the speech quality made by listeners. Listeners express their opinion on an absolute scale 1–5. This scale reflects the opinion on the quality of the synthesized speech or specificates what effort listeners have to spend during listening to the synthesized speech[21]. The scale is shown in the table 4.4.

The test is evaluated on the speech synthesized from the following text taken from a random czech internet page: *„Po horkých dnech Česko zkropí bouřky. Meteorologové varují, že v pátek odpoledne silné bouřky zasáhnou západ Čech. Postupně se budou přesouvat a večer se dají očekávat ve východní polovině Čech, a v noci na sobotu přejdou nad Českomoravskou vrchovinu a západní Moravu. Mohou je provázet přívalové srážky a kroupy, vítr dosáhne v nárazech až sedmdesát kilometrů za hodinu.“*

tuf	tur	tuš	tub	tul	tun
pyl	pih	pij	piš	piv	pin
pes	les	ves	bez	děs	rez
důl	hůl	vůl	sůl	půl	kůl
rak	tak	vak	sak	lak	pak
lev	les	lem	lep	led	len
byt	lid	kyt	žid	hit	vid
hole	pole	dole	role	kole	mole
loď	lom	lov	lok	los	lob
dub	dur	dul	duc	duch	duň
kos	bos	los	nos	šos	sos
bál	šál	tál	vál	sál	kál
pít	lít	být	žít	výt	mít
mok	mor	mol	moc	moč	mop
set	sen	sem	seč	sek	sel
pár	páv	pán	pád	pás	pák
kaz	ďas	pas	čas	tas	bas
val	kal	žal	dal	ťal	pal
kos	koš	kop	kol	kov	koj
pan	pal	pař	pas	pak	pac
suk	puk	kuk	luk	fuk	muk
mop	cop	sob	lob	top	zob
bod	Bob	bor	bol	bok	bos
jed	zet	med	ret	led	set
sok	dok	rok	šok	lok	bok
jam	ram	tam	lam	kam	dam
bič	bim	byl	bys	bych	byt
nes	neb	nech	neť	nej	než
her	per	ber	žer	ker	der
pech	cech	mech	dech	Čech	nech
sok	sos	sob	soch	sov	sod
dal	daň	dav	dam	dar	Dan
kus	bus	Rus	hus	pus	dus
lej	rej	dej	nej	sej	hej
mít	mís	mír	míň	míč	mým
věk	věž	vět	věn	Věr	věc
lež	než	veš	běž	jež	řeš
Fin	lin	čin	syn	pin	kin
rok	roh	rod	ros	roj	rom
kyt	kil	kyj	kyv	kin	Kim

Table 4.3: Groups of words selected for the MRT.

Rating	Speech quality	Listening effort
1	Bad	No meaning understood with any feasible effort
2	Poor	Considerable effort required
3	Fair	Moderate effort required
4	Good	Attention necessary, no appreciable effort required
5	Excellent	Complete relaxation possible, no effort required

Table 4.4: The scale of ratings for the MOS test.

4.5.3 Results of evaluation

Tests were evaluated in the form of online survey. Survey is included in the attachment C. Total of 24 participants took part in the evaluation composed mainly of students in age from 20 to 25 years. Results of MRT and MOS tests are shown in the tables 4.5 and 4.6.

Right answers	86 %

Table 4.5: MRT test evaluation results. Overall right answers value is averaged from partial results for each group of words.

Rating	Speech quality	Listening effort	Answered
1	Bad	No meaning understood with any feasible effort	0 %
2	Poor	Considerable effort required	4 %
3	Fair	Moderate effort required	48 %
4	Good	Attention necessary, no appreciable effort required	44 %
5	Excellent	Complete relaxation possible, no effort required	4 %

Table 4.6: MOS test evaluation results.

According to the survey evaluation, both MRT and MOS tests results are quite good. In MRT test only a few word groups resulted as unintelligible and the overall averaged right answers ration is high. In MOS test there is no *bad* answer and only a neglecting ratio of answers is *poor* and the majority of answers lie between *fair* and *good*. In compare, Festival-si Sinhala TTS system[28], have MRT test results 71%.

There is still space for improvement though. The resulting speech quality can be improved by using larger speech corpus or implementing and improving TTS component modules in the MARY system. Better text preprocessing could also help because in the current state only a simple interpunction examination (i.e. pauses according to interpunction and end of sentences detection) and POS tagging is done.

4.6 Results

As a result of adding the new language module and the new voice to the MARY system several simple programs and scripts to ease database creation and to ease system integration were created and the whole process was described step by step. The database corpus was created and the speech database was recorded. The new language module and the new voice were successfully integrated into the MARY system and released as a candidate for being published in the next official MARY system version (`https://github.com/marytts/marytts/pull/184`). The successful result was shown with the demonstrator programs. The quality of the synthesized speech was evaluated via online survey and discussed.

Chapter 5

Grapheme to phoneme transcription

This chapter describes the implementation of the own grapheme-to-phoneme (GTP) module.

5.1 GTP module

The GTP description and Czech phonetics was described in chapter 2. This chapter describes implementing such a module for Czech. GTP module is designed as a pure rule-based GTP. There is no dictionary neither for foreign nor native words. This solution was chosen because this work is focused on Czech.

In the following sections, the design, implementation and testing are described.

5.1.1 Czech GTP rules

Czech is appropriate for using transcription rules because of a strong relation between spelling and pronunciation. Therefore the count of rules is quite low.

For the following description the Czech SAMPA[16] is used. During the GTP implementation several groups of rules were used:

- **Basic rules** – These rules are used for every transcription regardless the context. The exapmle is: *ch ->x*.
- **Vowel connection** – The transcription differs if two neighbour vowels are part of the same syllable. If yes, diftong rules are used, e.g. *ou ->ou*
- **Consonant and vowel connection** – The only exceptions in this group are connecion of [d, t, n] with [i] (e.g. rule *d ->D*) and the letter ě in connection with [b, p, v] (rule *ě ->je*).
- **Sonority assimilation** – There are two types of assimilation:
  - Regressive assimilation – e.g. *shoda - [zhoda]*, this type is prevailing.
  - Progressive assimilation – e.g. *shoda - [schoda]*.

The other graphemes are transcripted right into their phoneme equivalent (e.g. *ř ->R*)Using these groups of rules the GTP transcription can be successfully implemented.

5.1.2 Syllabification task

The syllabification is the process of segmentation of the words to syllables. The syllabification has a big influence on the prosody generation and therefore the right syllabification is an important task during the TTS synthesis.

Syllables are basic sound units to which phonemes are grouped together. Syllables consist of **syllable nucleus** (most often vowels) and **margins** (typically consonants). However in Czech also syllabic consonants can make up the syllabic nucleus. Such consonants are: *l, r, m, n* and they became syllabic consonants in neighbourhood of other consonants.

The syllabification is based on syllable nucleus and margins word structure. One of the approaches to syllabification can be usage of N-Grams[18]. The multigram model assumes that language can be described as the output of a memoryless source that emits variable-length sequences of words. The estimation of the model parameters can be formulated as a Maximum Likelihood estimation problem from incomplete data.

For the phonemes classification the sonority scale can be used[17]. The sonority scale is a list of phonetic segments showing the relative resonance of phonetic segments in relation to other segments. According to this scale phonemes can be classified. The syllable nucleus will have the highest weight and in syllable only one scale peak can be contained. Defined sonority scale for this project is showed in the table 5.1.

Phoneme group	Weight
[a]	10
[e, o]	9
[i, u]	8
[r]	7
[l]	6
[m, n]	5
[z, v, Z]	4
[s, f, j, S, N, ch]	3
[b, d, g, D, c, C]	2
[p, t, k, T]	1

Table 5.1: Sonority scale defined for this project.

5.1.3 Implementation

The GTP method was implemented in Java as a part of the demonstrator application. It implements the rules described above as a set of conditions. Every sentence is tokenized to words, and each word is examined stand-alone. The word is iterated character by character from the end to the start. The reverse iteration allows effectively solve the multiple sonancy assimilation problem[19]. During iteration the rules are applied and the GTP transcription processed consecutively.

The syllabification part was not successfully implemented in the application due to the algorithm implementation problems.

5.1.4 Results

The implemented GTP method was evaluated in comparison to the GTP module trained for the MARY system during the process of the new language creation. The text database created for the speech database recording was selected as a basis for testing dataset.

The text database needed to be transformed to fit the purpose of the test. Due to the lack of the text preprocessing module in our GTP method, all non-alphabetic symbols were erased from the text. The resulting dataset contains only words composed of alphabetic characters separated by one space. This dataset is now suitable for the testing.

For testing the pure GTP without syllabifying, transcription results from the MARY GTP had to be transformed too. All syllable marking symbols were erased.

As a testing method the character difference was selected. Both output files were compared characted by character and differences were stored. The test results showed many differences between the output files. This can be caused by the fact that implemented GTP process only isolated words and some rules need context of another words. Another reason can be that implemented GTP strictly follows the rules of the literary Czech while MARY GTP is trained on the dictionary which was created semi-automatically. Both semi-automatic creation and GTP training could bring transcritpion mistakes to the GTP model. Examples of different transcription of words are shown in the table 5.2.

Implemented GTP	MARY GTP
f T i p	v T i p
l e c g d o	l e dz g d o
s j e z t	s j e s t
o p r a f t e	o p r a v t e
j e z e f C i:	j e z e v C i:

Table 5.2: Examples of different transcription between tested GTP modules.

The results of GTP module didn't meet the results of the MARY GTP module. However this do not need necessary mean that the functionality is wrong because MARY GTP model can contain some mistakes or rules which do not follow the literary Czech. Syllabifying part of module was not successfully implemented so the output of the module is only sequence of phonemes. For further improvements syllabification algorithm could be implemented and tested with the MARY TTS system.

Chapter 6

Conclusion

This work deals with text-to-speech(TTS) and provides a general theoretical introduction to TTS systems. The components of the TTS systems were studied and most widely used methods were described. The MARY TTS system architecture was inspected and described and the process of adding the new language module and the new HMM-based voice to the MARY TTS system was explained.

The practical result of this work is created Czech language module and Czech HMM-based voice for the MARY TTS system. These components were successfully integrated into the MARY TTS system, tested using implemented demonstrator application and evaluated. Both components were published for the new release of the MARY TTS system.

Another contribution of this project is the step-by-step tutorial of creating the new language module and the new voice which can be used for repeating experiment in another language or with different data set. Several complementory programs were implemented including audio recorder application which can ease this procedure. Also the created speech database can be used for further work.

This work have potencial for several future improvements. The language module can be further improved by reimplementing TTS components and tuning them for Czech. To improve the synthesized speech quality also larger database for voice creation can be used. Another scope of the future work can be focused on the MARY TTS system. Automated language module creation from minimal NLP components or better connection between MARY TTS system and MySQL database could be implemented and publiblished for the future releases.

Bibliography

[1] Czech graphemes frequencies.
http://www.czech-language.cz/alphabet/alph-prehled.html.

[2] GIT extensions. https://code.google.com/p/gitextensions/.

[3] Hmm voice creation.
https://github.com/marytts/marytts/wiki/HMMVoiceCreation.

[4] HTK toolkit. http://htk.eng.cam.ac.uk/.

[5] HTS toolkit. http://hts.sp.nitech.ac.jp/.

[6] Libcurl - the multiprotocol file transfer library. http://curl.haxx.se/libcurl/.

[7] Linguistic/prosodic processing.
http://festvox.org/festtut/notes/festtut_6.html.

[8] MARY TTS system. https://github.com/marytts/marytts/wiki/MARY-TTS-5.0.

[9] Maven - welcome to Apache Maven. http://maven.apache.org/.

[10] Minimalist GNU for windows. http://www.mingw.org/.

[11] MySQL :: The world's most popular open source database. http://www.mysql.com/.

[12] Netbeans IDE – the smarter and faster way to code. https://netbeans.org/.

[13] New language support.
https://github.com/marytts/marytts/wiki/New-Language-Support.

[14] OpenAL soft library. http://kcat.strangesoft.net/openal.html.

[15] QT toolkit. http://qt-project.org/.

[16] SAMPA for Czech. http://www.phon.ucl.ac.uk/home/sampa/czech-uni.htm/.

[17] What is the sonority scale? http://www-01.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsTheSonorityScale.htm.

[18] S. Deligne and F. Bimbot. Language modeling by variable length sequences : Theoretical formulation and evaluation of multigrams. In *In Proc. ICASSP*, pages 169–172, 1995.

[19] J. Psutka and L. Müller and J. Matoušek and V. Radová. *Mluvíme s počítačem česky*. Academia, 2006. ISBN 80-200-1309-1.

[20] L. F. Lamel, R. Kassel, and S. Seneff. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *In Proceedings of the DARPA Speech Recognition Workshop*, pages 100–110, 1986.

[21] P. A. Naylor and N. D. Gaubitch. *Speech Dereverberation.* Springer, 2010. ISBN 978-1-84996-055-7.

[22] E. Schlunz, G.I. Barnard and G.B. Van Huyssteen. Part-of-speech effects on text-to-speech synthesis. In *21st Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, pages 257–262, 2010.

[23] M. Schröder and J. Trouvain. The german text-to-speech synthesis system MARY: A tool for research, development and teaching. *International journal of speech technology*, 6:365–377, 2003.

[24] T. Dutoit. *An introduction to text-to-speech synthesis.* Kluwer Academic Publishers Norwell, 1997. ISBN 0-7923-4498-7.

[25] D. Tihelka, J. Kala, and J. Matoušek. Enhancements of viterbi search for fast unit selection synthesis. In *Proceedings of Int. Conf. Interspeech 2010*, pages 174–177, 2010.

[26] D. Tihelka and J. Matoušek. The design of Czech language formal listening tests for the evaluation of TTS systems. In *LREC 2004, proceedings of 4th International Conference on Language Resources and Evaluation*, pages 2099–2102, 2004.

[27] D. Tihelka and J. Romportl. Exploring automatic similarity measures for unit selection tuning. In *Interspeech 2009, proceedings of 10th Annual Conference of International Speech Communication Association*, pages 736–739, 2009.

[28] R. Weerasinghe, A. Wasala, V. Welgama, and K. Gamage. Festival-si: A sinhala text-to-speech system. In *Lecture Notes in Computer Science, Volume 4629*, pages 472–479, 2007.

[29] T. Yoshimura. *Simultaneous modelling of phonetic and prosodic parameters, and characteristic conversion for HMM based text-to-speech systems.* PhD thesis, Department of Electrical and Computer Engineering Nagoya Institute of Technology, 2002.

[30] Z. Palková. *Fonetika a fonologie češtiny.* Karolinum, 1994. ISBN 8070668431.

Appendix A

CD Contents

Apps – directory containing applications implemented in this project
MARY – directory containing MARY Czech language module and Czech HMM voice
Speech database – directory containing text and speech data of the database
Technical report – directory containing technical report document with sources

Appendix B

Czech allophones set

Listing B.1: Phone set in the XML format for the MARY system for the Czech language. Articulation features for vowels: Vlng marks if the vowel is short, long or diphtong, vheight and vfront marks tongue position and vrnd marks lips roundness. For consonants: ctype is manner of production(e.g fricative), cplace is place of production(e.g. labial) and cvox marks if consonant is voiced.

```
<allophones name="sampa" xml:lang="cs" features="vlng vheight vfront
vrnd ctype cplace cvox">
  <silence ph="_"/>

  <vowel ph="i" vlng="s" vheight="3" vfront="3" vrnd="+"/>
  <vowel ph="e" vlng="s" vheight="2" vfront="3" vrnd="+"/>
  <vowel ph="a" vlng="s" vheight="1" vfront="2" vrnd="+"/>
  <vowel ph="o" vlng="s" vheight="2" vfront="1" vrnd="+"/>
  <vowel ph="u" vlng="s" vheight="3" vfront="1" vrnd="+"/>
  <vowel ph="i:" vlng="l" vheight="3" vfront="3" vrnd="+"/>
  <vowel ph="e:" vlng="l" vheight="2" vfront="3" vrnd="+"/>
  <vowel ph="a:" vlng="l" vheight="1" vfront="2" vrnd="+"/>
  <vowel ph="o:" vlng="l" vheight="2" vfront="1" vrnd="+"/>
  <vowel ph="u:" vlng="l" vheight="3" vfront="1" vrnd="+"/>
  <vowel ph="ou" vlng="d" vheight="3" vfront="1" vrnd="+"/>
  <vowel ph="au" vlng="d" vheight="3" vfront="1" vrnd="+"/>
  <vowel ph="eu" vlng="d" vheight="3" vfront="1" vrnd="+"/>
  <vowel ph="@" vlng="a" vheight="2" vfront="2" vrnd="-"/>

  <consonant ph="p" ctype="s" cplace="l" cvox="-"/>
  <consonant ph="b" ctype="s" cplace="l" cvox="+"/>
  <consonant ph="t" ctype="s" cplace="a" cvox="-"/>
  <consonant ph="d" ctype="s" cplace="a" cvox="+"/>
  <consonant ph="c" ctype="s" cplace="a" cvox="-"/>
  <consonant ph="k" ctype="s" cplace="p" cvox="-"/>
  <consonant ph="g" ctype="s" cplace="p" cvox="+"/>
  <consonant ph="T" ctype="s" cplace="p" cvox="-"/>
  <consonant ph="D" ctype="s" cplace="p" cvox="+"/>
  <consonant ph="ts" ctype="a" cplace="a" cvox="-"/>
  <consonant ph="dz" ctype="a" cplace="a" cvox="+"/>
  <consonant ph="tS" ctype="a" cplace="a" cvox="-"/>
  <consonant ph="dZ" ctype="a" cplace="a" cvox="+"/>
  <consonant ph="C" ctype="a" cplace="a" cvox="-"/>
  <consonant ph="f" ctype="f" cplace="l" cvox="-"/>
  <consonant ph="v" ctype="f" cplace="l" cvox="+"/>
  <consonant ph="s" ctype="f" cplace="a" cvox="-"/>
  <consonant ph="z" ctype="f" cplace="a" cvox="+"/>
  <consonant ph="S" ctype="f" cplace="a" cvox="-"/>
  <consonant ph="Z" ctype="f" cplace="a" cvox="+"/>
  <consonant ph="j" ctype="f" cplace="p" cvox="+"/>
  <consonant ph="x" ctype="f" cplace="p" cvox="-"/>
  <consonant ph="h" ctype="f" cplace="g" cvox="+"/>
  <consonant ph="R" ctype="f" cplace="a" cvox="+"/>
  <consonant ph="r" ctype="l" cplace="a" cvox="+"/>
  <consonant ph="l" ctype="l" cplace="a" cvox="+"/>
  <consonant ph="m" ctype="n" cplace="l" cvox="+"/>
  <consonant ph="n" ctype="n" cplace="a" cvox="+"/>
  <consonant ph="N" ctype="n" cplace="p" cvox="+"/>

</allophones>
```

Appendix C

Speech evaluation survey

Speech quality evaluation

Dear Sir / Madam,

thank you for visiting us. By filling out this 5 minute survey, you will help us obtain the very best results.

Rank synthesized speech according to quality and listening effort

Video: https://www.youtube.com/embed/1nrs21T30Xo

- ◯ Speech quality: Bad, Listening effort: No meaning understood with any feasible effort
- ◯ Speech quality: Poor, Listening effort: Considerable effort required
- ◯ Speech quality: Fair, Listening effort: Moderate effort required
- ◯ Speech quality: Good, Listening effort: Attention necessary, no appreciable effort required
- ◯ Speech quality: Excellent, Listening effort: Complete relaxation possible, no effort required

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/J0X-u3mJkiU

- ◯ tuf
- ◯ tur
- ◯ tuš
- ◯ tub
- ◯ tul
- ◯ tun

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/UVxP3nGfwSI

- ◯ pyl
- ◯ pih
- ◯ pij
- ◯ piš
- ◯ piv
- ◯ pin

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/SUVcNKo1p6I

- ◯ pes
- ◯ les
- ◯ ves
- ◯ bez
- ◯ děs
- ◯ rez

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/Wj2TswO649O

- ◯ důl
- ◯ hůl
- ◯ vůl
- ◯ sůl
- ◯ půl
- ◯ kůl

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/jZY-ZEl3kKM

- ○ rak
- ○ tak
- ○ vak
- ○ sak
- ○ lak
- ○ pak

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/lDSkjS1lskw

- ○ lev
- ○ les
- ○ lem
- ○ lep
- ○ led
- ○ len

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/F0Y8lv3qLEM

- ○ byt
- ○ lid
- ○ kyt
- ○ žid
- ○ hit
- ○ vid

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/uk-emUpDL2U

- ◯ hole
- ◯ pole
- ◯ dole
- ◯ role
- ◯ kole
- ◯ mole

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/qbbjIhL0QU8

- ◯ loď
- ◯ lom
- ◯ lov
- ◯ lok
- ◯ los
- ◯ lob

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/ma5LiDhDd5E

- ◯ dub
- ◯ dur
- ◯ dul
- ◯ duc
- ◯ duch
- ◯ duň

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/_BzxIYUIj8o

- ○ kos
- ○ bos
- ○ los
- ○ nos
- ○ šos
- ○ sos

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/SMopMIpQ7CI

- ○ bál
- ○ šál
- ○ tál
- ○ vál
- ○ sál
- ○ kál

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/46wtap_8SO0

- ○ pít
- ○ lít
- ○ být
- ○ žít
- ○ výt
- ○ mít

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/tzRfrtCD8xw

- ○ mok
- ○ mor
- ○ mol
- ○ moc
- ○ moč
- ○ mop

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/SHgwpl_k3oc

- ○ set
- ○ sen
- ○ sem
- ○ seč
- ○ sek
- ○ sel

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/Rjp8zyKANVw

- ○ pár
- ○ páv
- ○ pán
- ○ pád
- ○ pás
- ○ pák

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/YYAL_mjR7tM

- ◯ kaz
- ◯ ďas
- ◯ pas
- ◯ čas
- ◯ tas
- ◯ bas

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/-OIs_zDH34o

- ◯ val
- ◯ kal
- ◯ žal
- ◯ dal
- ◯ ťal
- ◯ pal

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/CZ26FY1YVac

- ◯ kos
- ◯ koš
- ◯ kop
- ◯ kol
- ◯ kov
- ◯ koj

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/07N7RH2X20g

- ◯ pan
- ◯ pal
- ◯ pař
- ◯ pas
- ◯ pak
- ◯ pac

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/YBEWd-EUFj4

- ◯ suk
- ◯ puk
- ◯ kuk
- ◯ luk
- ◯ fuk
- ◯ muk

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/tzObererjyE

- ◯ mop
- ◯ cop
- ◯ sob
- ◯ lob
- ◯ top
- ◯ zob

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/tmE-vx7PUAA

- ◯ bod
- ◯ Bob
- ◯ bor
- ◯ bol
- ◯ bok
- ◯ bos

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/mKdn3IXtAEM

- ◯ jed
- ◯ zet
- ◯ med
- ◯ ret
- ◯ led
- ◯ set

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/cUetOMgdfj0

- ◯ sok
- ◯ dok
- ◯ rok
- ◯ šok
- ◯ lok
- ◯ bok

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/ncNe5bGtTDo

- ◯ jam
- ◯ ram
- ◯ tam
- ◯ lam
- ◯ kam
- ◯ dam

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/OtT4GFXrkiA

- ◯ bič
- ◯ bim
- ◯ byl
- ◯ bys
- ◯ bych
- ◯ byt

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/atzxcn2VoXk

- ◯ nes
- ◯ neb
- ◯ nech
- ◯ neť
- ◯ nej
- ◯ než

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/gW90PGK2ruQ

- ◯ her
- ◯ per
- ◯ ber
- ◯ žer
- ◯ ker
- ◯ der

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/NcwdPFjJK4k

- ◯ pech
- ◯ cech
- ◯ mech
- ◯ dech
- ◯ Čech
- ◯ nech

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/fd3B1-R90IM

- ◯ sok
- ◯ sos
- ◯ sob
- ◯ soch
- ◯ sov
- ◯ sod

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/Y3U_8sCuG3M

○ dal

○ daň

○ dav

○ dam

○ dar

○ Dan

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/FBgHy1Nh7v8

○ kus

○ bus

○ Rus

○ hus

○ pus

○ dus

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/3XsILhxYlf4

○ lej

○ rej

○ dej

○ nej

○ sej

○ hej

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/xzU1Mjjary4

- ◯ mít
- ◯ mís
- ◯ mír
- ◯ míň
- ◯ míč
- ◯ mým

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/pZ6OFioLv5s

- ◯ věk
- ◯ vež
- ◯ vět
- ◯ věn
- ◯ Věr
- ◯ věc

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/mSPhpGqx31I

- ◯ lež
- ◯ než
- ◯ veš
- ◯ běž
- ◯ jež
- ◯ řeš

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/EiBaw3-resM

- Fin
- lin
- čin
- syn
- pin
- kin

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/lZvgyXWXKLs

- rok
- roh
- rod
- ros
- roj
- rom

Decide which word is contained in speech sample

Video: https://www.youtube.com/embed/PFfLLd_4KXk

- kyt
- kil
- kyj
- kyv
- kin
- Kim