Gudeta Gebremariam

Speech synthesis

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

23.03.2016

Author(s) Title Number of Pages Date	Gudeta Gebremariam Speech synthesis 35 pages + 1 appendices Wednesday 23rd March, 2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Olli Hämäläinen, Senior Lecturer

Speech is the most natural media of communication among the human being. Speech synthesis system has been implemented in computer systems for various purposes like accessibility, education and communication aid in mass transit.

The main goal of this thesis is to study the technology behind speech synthesis followed by comparison of existing services and an implementation of speech technology in an application intended to assist new language learning.

Keywords	Language, Speech Synthesis, Speech, Web application

Contents

Abbreviations

API	Application Programming Interface
ASR	Automated Speech Recognition
CALL	Computer Assisted Language Learning
HMM	Hidden Markov Model
HTK	Hidden Markov Toolkit
HTTP	Hyper Text Transfer Protocol
LAMP	Linux, Apache, MySQL and PHP
SSML	Speech Synthesis Markup Language
TTS	text-to-speech
W3C	World Wide Web Consortium

1 Introduction

The process in which a computer is used to produce speech is known as speech synthesis. There are multiple applications of speech synthesis and one among them is education. When a computer is used to assist a learner to improve language skills, it is known as computer assisted language learning (CALL).

The main objective of this project is to design a language learning tool that implements speech technology. In the application, an existing speech synthesis technology is used. In the process, Finnish language speech synthesizers were compared first and then a web-based language learning tool was implemented where the target user is a new language learner. In the application, the user is able to load a text of his/her interest and able to read it aloud using a speech synthesizer.

This thesis is organized in-to six chapters. After this introduction, the theoretical background of the speech synthesis technology is discussed. The historical evolution of speech synthesis systems along with the technology behind various types of speech synthesis methods and computer-assisted language learning will be covered. Also the phonetic characteristics of the Finnish language and the application areas of synthesized speech will be discussed. In chapter three, current available speech synthesis technologies on the market are covered. Then chapter four focuses on some current research on speech technologies such as visualization of speech and automatic speech recognition. Also emerging markets for the technology are covered. Finally, chapter five discusses the implementation of a language learning application that uses speech technologies.

2 Theoretical background

Language is the main mechanism of communication among human beings. In human communication, speech is specialized for rapid transfer of information [1]. Articulatory phonetics is a subfield of phonetics that studies how humans produce speech sounds through the interaction of different physiological structures. Naturally a human being synthesizes voice using the vocal tract and lungs act as a power source. Air flows through the trachea and the lips, tongue, jaw and velum are used as articulators. [2,14].

Figure 1 shows the human articulatory system. Consonant sounds are produced with some restriction or closing of a vocal tract that restricts air flow from the lungs. This is also called the place of articulation. On the other hand a vowel is produced with an air flow through an open vocal tract.

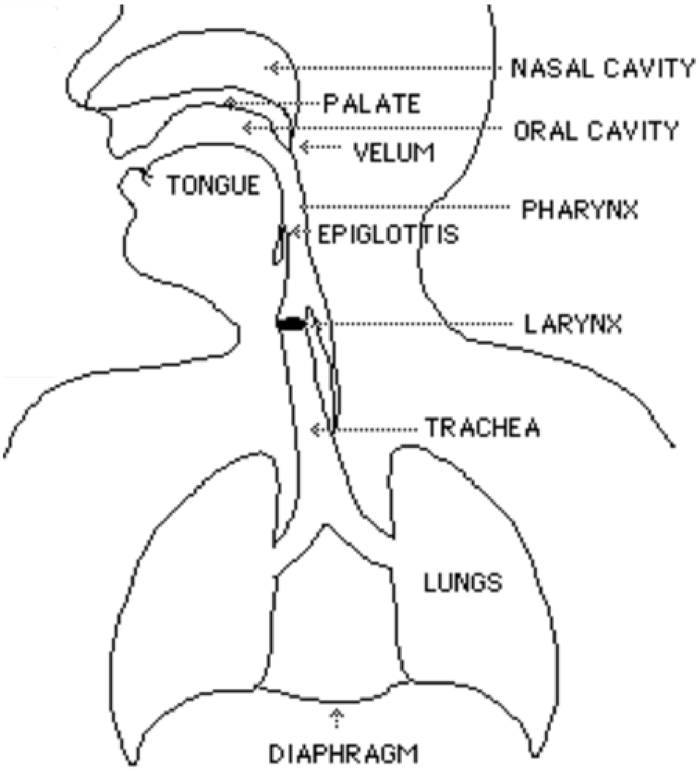


Figure 1: The human speech production system. Copied from [1].

Speech synthesis is the artificial production of human speech using a computer, either hardware or software. A text-to-speech synthesis system is a system that converts written text-to-speech. There are two main steps in text-to-speech-synthesis. These are known as text analysis and speech waveform generation. In the text analysis phase, an input text is converted into phonetic or other linguistic format. In the second step an audio output is formed. Figure 2 shows this operation. [2,2].

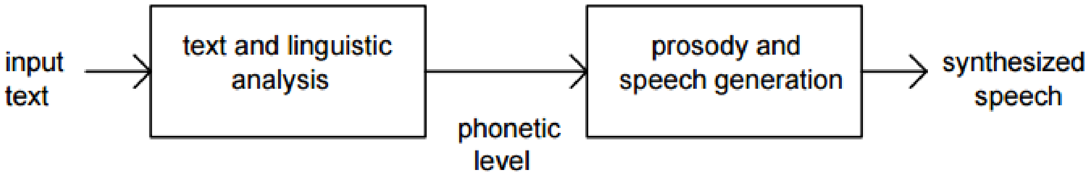


Figure 2: Simple text-to-speech synthesis procedure. Copied from [2].

The main objective of this theoretical background section is to examine the historical background, current status, architecture and design process of speech synthesizers. The materials for the literature review are mainly collected from on-line sources. Later the review will be used in an implementation of an existing speech synthesis system in developing an application targeted at assisting language learning.

2.1 Brief history of speech synthesis

The first attempts to build artificial speech were carried out using mechanical devices to produce vowels. Later in the 1920s the first electrical synthesizer was introduced by Stewart. The synthesizer was capable of generating sounds of single static vowels. In 1939 VODER was introduced in the New York World fair. It was seen as the first speech sythesizer. In 1953 the Parametric Artificial Talker (PAT) was introduced and it was considered as the first formant synthesizer. It had three formant resonators connected in parallel. The first integrated circuit speech synthesis system is perhaps the Votrax chip. In 1980, Texas Instruments introduced a product based on a low cost TMS-5100 chip called Speak-n-Spell which was designed to be a reading aid to children. Modern speech synthesis systems employ more sophisticated algorithms such as hidden Markov models (HMM) and neural networks. [2, 4-10.]

Text-to-speech systems are now employed in a variety of applications including assistive tools, education, telecommunication, entertainment, navigation guidance, improving literacy [3] and as an alternative way for human-computer interaction (HCI). Clarity and naturalness are used to rate the quality of a speech synthesis system. [2, 79.]

2.2 Speech synthesis techniques

Currently one of the main methods used to create a speech synthesis system is by concatenating recorded speech units like phones or diphones. Sometimes the speech units could even be words and sentences for specific usage domains. [4]. Another methodology commonly known as formant synthesis or rule-based synthesis uses an additive synthesis and acoustic model. In a formant synthesizer the parameters of an input signal are varied to form artificial speech. [5]. In addition to the above mentioned methods, Articulatory synthesis is a method that simulates human vocal tract and HMM-based synthesis implements Hidden Markov Model algorithms in speech synthesis [2].

The Source-filter-model of speech is used as a basis for formant synthesis. The formant-based synthesis technique is regarded more flexible because an infinite number of sounds could be produced. Intelligible speech requires at least three formants. Every formant is modeled with two pole resonators. This enables the formant frequency as well as bandwidth to be specified. A cascade formant synthesizer comprises band-pass resonators which are connected to one another in series [2] as shown in figure 3.

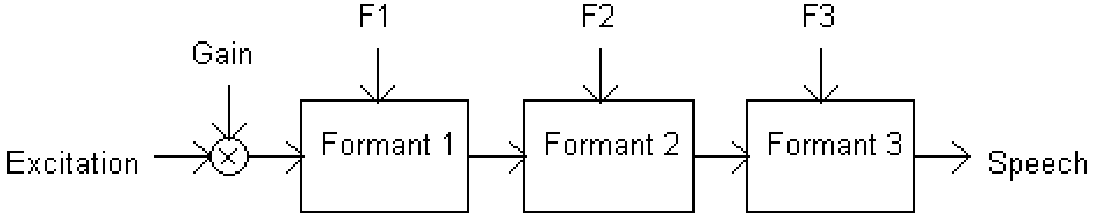


Figure 3: Cascade formant synthesizer. Copied from [2].

ESpeak is an open-source software implementation of a formant type synthesizer. It can use either its own espeak engine or Klatt. Voiced speech sounds such as vowels and sonorant consonants are created by adding together sine waves to make the formant peaks. Unvoiced consonants are made by playing recorded sounds. Voiced consonants are made by mixing a synthesized voice with a recorded unvoiced sound. Klatt uses the same formant data but produces voiced sounds by starting with a waveform and applying digital filters. [6]

The Unit selection method uses large recorded speech data sufficient to cover language features and then speech is produced by cutting and stitching units from a database [7]. The concatenation method depends mainly on runtime selection and compilation of speech units from the database. For example, according to Acapela Group speech synthesis process for the word **impressive** is made up from chunks from the words "impossible", "president" and "detective", which results in a natural sounding word. According to Silen et al. (2007), speech synthesis based on unit selection can produce more natural sounding voice than other methods [8, 1] Contrary to concatenative systems, formant-based synthesis does not initially use a recorded human voice and as a result the output sound is more robot-sounding. However the main advantage of these systems is a smaller footprint of the programs due to lack of a large database. Additionally the output voice does not suffer from glitches and a faster output could be produced. [4]

A development of a unit selection speech synthesizer developed for the Finnish language at the Tampere University of Technology (TUT) illustrates the process of the unit selection concatenative synthesizer. This method uses a large pre-recorded speech inventory so that a sufficient phonetic and prosodic coverage for a language is provided. As the name indicates, units are cut and concatenated from a database to produce speech. Unit selection from the database is guided by two costs, target and join cost. The former measures the similarity of the candidate and desired units. On the other hand, the latter measures the concatenation quality of two consecutive units. The design of the database is important and should consider the target language because the quality of synthesized speech highly depends on the coverage of the database. [8]

To design a unit selection speech synthesizer, first and foremost, understanding of the target language is important even though a unit selection synthesizer could be built with no or little knowledge of the target language [8]. Hence the authors, who developed the unit selection Finnish speech synthesizer at TUT, studied the phoneme and orthography of the Finnish language and what makes it similar to and different from other languages. Secondly a database was designed to store phonemes from a recorded speech units. Next, a synthesis engine was implemented and tested. [8]

A good example that implements diphone-based synthesis is the MBROLA project. A diphone is an adjacent pair of phones. Building new voices for MBROLA is efficient using diphone and sentence extraction tools. A Diphone database is a diphone synthesizer and it has to include all possible diphones in the language it is designed for. The general procedure of MBROLA voice creation is as follows. [9]

The first step is **creating a text corpus**. Here a list of the phones including allphones for a given language is prepared followed by a list of diphones. Then a list of words containing all the diphones is created and each diphone should appear at least once. This is followed by putting keywords in a career sentence. The next step is **recording the corpus**. The corpus is read by a professional speaker with monotonous intonation and the speech is digitally recorded and stored in a digital format. After this the **corpus is segmented**. Here the diphones must be found and annotated and then the position of the border between the phones is marked. Finally **leveling** is done. Here the energy levels at the beginning and end of a segment are averaged and the pitch is normalized. After these steps diphone files are saved in a WAV format along with the diphone database file. [9]

2.3 Computer-assisted language learning (CALL)

According to Philip Hubbard, computer-assisted language learning is defined as any process in which a learner uses a computer and as a result improves his or her language. It is a broad and dynamic field of study. Here, by computer it means in its broadest sense, including networks and other portable devices. By improving it means from various perspectives such as learning effectiveness, access to various learning materials, convenience across a wide range of time and place, motivational aspects and less or fewer expensive resource requirements. [10]

With regard to learning skills such as listening, speaking and pronunciation, computers and networks have evolved so that a large amount of information is readily available. These days listeners have access to a wealth of audio and video materials to listen to usually for free. In the future of language learning, the primary question will be how users choose materials that match their learning styles rather than which programs offer what kind of features. Speaking exercises could be accomplished via a user talking to each other through an IT medium or users recording their sounds and playing them back. Automatic speech recognition systems could also be utilized. Pronunciation learning could be implemented where students could try to match their pronunciation relative to the one produced by a native speaker. Graphic representations of speech as a waveform, for example, could help to compare the sounds produced by a student to a native speaker. [10]

Computer programs could assist reading and writing in multiple ways. In addition to the availability of multiple print materials in an accessible way, collaborative writing methods such as Google Docs have provided a way to write and share at the same time. Spelling, grammar and thesaurus checkers and on-line dictionaries also fit into the context of reading and writing with CALL. [10]

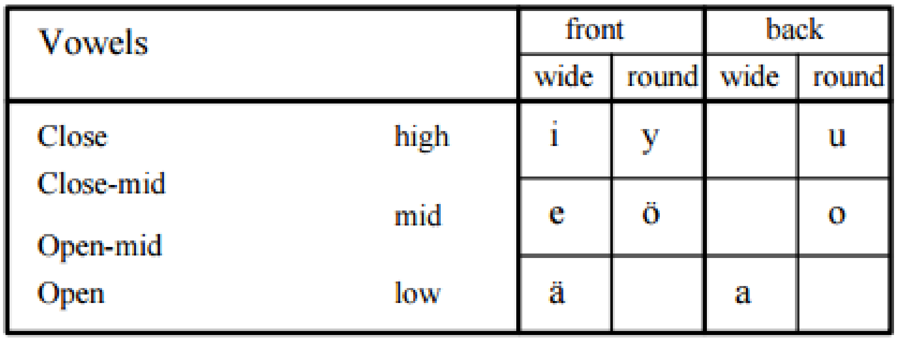
2.4 Language learning and technology

A vast amount of materials is accessible on-line for learning various languages. An informal form of learning could take many kinds of approaches. As an example, gaming and other fun ways of on-line collaboration are one form of informal learning. Additionally, peer to peer networks on various subjects and on-line dictionaries including translation services can be used by learners in developing their skills. For developing skills in vocabulary, there are programs like Quizlet and other flash card applications. Furthermore, for a longer term vocabulary retention, there exist advanced algorithms that help students to maintain an optimal rhythm through time. Pod-casts and various video materials help in advancing listening comprehension.

While the amount of information for an independent learner could be overwhelming, one can personalize his/her individual experience. There is a great potential in using personalized on-line resources. Mobile phones have brought high level of personalization in that they can be used in and out of class and information could be looked up with out restriction of place.

2.5 Finnish phonetics and phonology

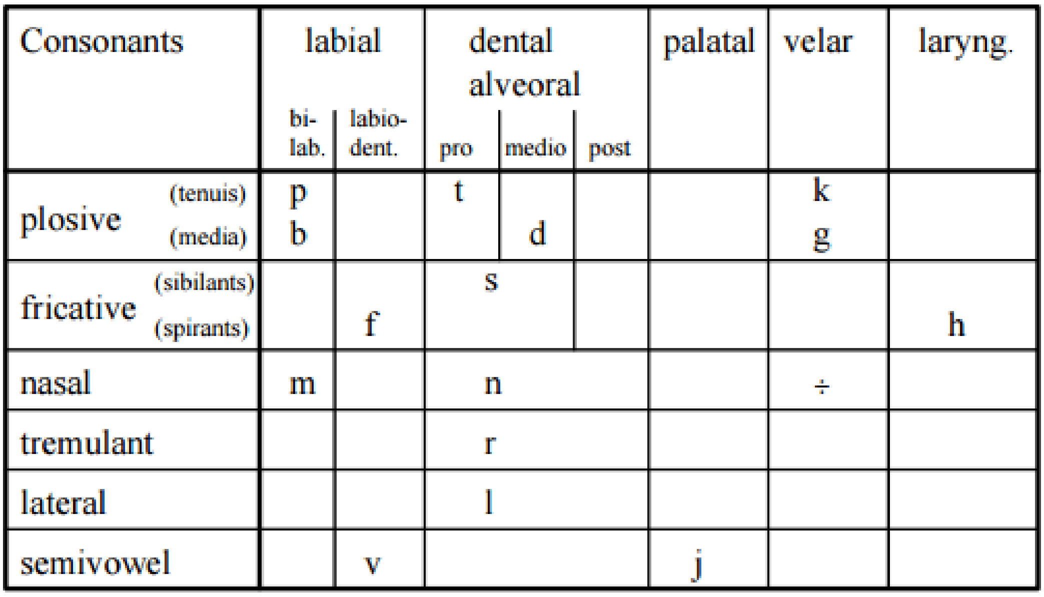
While studying speech technologies, it is important to know the characteristics of the target language. A phoneme is the smallest unit of speech that can be used to make one word different from another word. Phonemes are divided into consonants and vowels. In the Finnish language there exist eight vowels. These are /ɑ/, /e/, /i/, /o/, /u/, /y/, /æ/ and /ø/. Vowels can occur in both a short and long form and sequences and diphthongs, which are adjacent vowel sounds occurring within the same syllable. [8] Figure 4 summarizes the categorization of Finnish vowels. The Finnish orthography is phonemic: each phoneme corresponds to a certain grapheme (the smallest unit in a writing system). One or more syllables exist in every word. Each syllable in Finnish has a vowel as a sonant, which means that every word has at least a one vowel. [8]



Vowels		front		back	
		wide	round	wide	round
Close	high	i	y		u
Close-mid Open-mid	mid	e	ö		o
Open	low	ä		a	

Figure 4: Classification of Finnish vowels. Copied from [2].

Depending upon the place of articulation consonants in Finnish are classified as plosives (the vocal tract is closed causing a stop), fricatives (the vocal tract is tightened over some spot that the turbulent air creates sound), nasals (the vocal tract is closed; however the velum opens a course to the nasal cavity), tremulants (the top of the tongue is vibrating), laternals (the top part of the tongue shuts the vocal tract) and semivowels (almost like vowels but unstable). Figure 5 summarizes the Finnish consonants. [2]



Consonants		labial		dental alveoral			palatal	velar	laryng.
		bi-lab.	labio-dent.	pro	medio	post			
plosive	(tenuis)	p		t				k	
	(media)	b			d			g	
fricative	(sibilants)			s					
	(spirants)		f						h
nasal		m		n				÷	
tremulant				r					
lateral				l					
semivowel			v				j		

Figure 5: Classification of Finnish consonants. Copied from [2].

2.6 Natural language processing

Natural language processing (NLP) is a field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and human languages. The main challenges of the field include natural language understanding and natural language generation.

Natural language processing in the context of speech synthesis comprises various elements. Figure 7 illustrates the role of natural language processing in a unit selection TTS system. As shown in the figure, when a given text is fed into the parsing and structural analysis module, the text boundaries are identified. These identified sentences are then fed into normalization module. Here numbers, addresses, acronyms and other special tokens are identified and expanded and converted into spoken forms. Figure 6 shows basic text processing. The text processing phase includes text-to-phoneme conversion, which is undertaken based on syllable-to-phoneme conversion rules. [7]

The normalization module depends on lexicons and rule-based approaches. The prosody module determines the pitch contour, duration and intensity of the text to be synthesized. In general the NLP module is responsible for parsing, analyzing and transforming the input

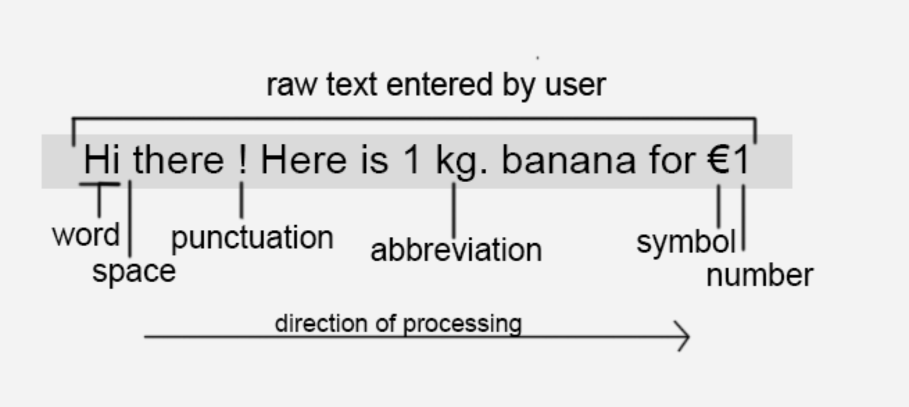


Figure 6: Raw text processing.

text into an intermediate symbolic format suitable to be processed using the DSP (Digital Signal Processing) module. [7].

Natural language processing systems are widely used in commercial language applications. Apple's Siri and Google now use these technologies. Deep neural networks implement large datasets and have great potential for use in language learning. [21]

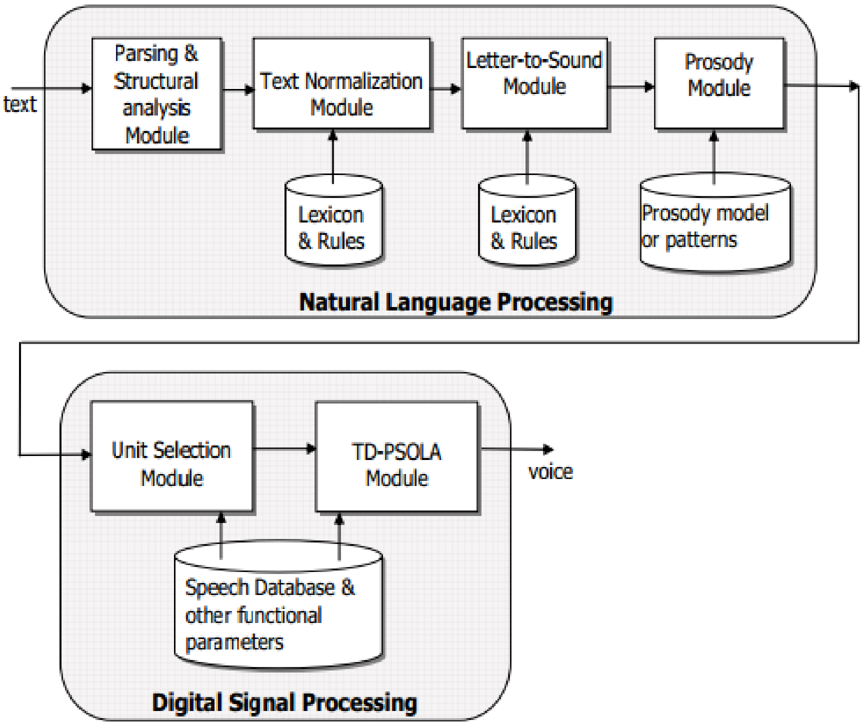

text
Parsing & Structural analysis Module
Text Normalization Module
Letter-to-Sound Module
Prosody Module
Lexicon & Rules
Lexicon & Rules
Prosody model or patterns
Natural Language Processing
Unit Selection Module
TD-PSOLA Module
voice
Speech Database & other functional parameters
Digital Signal Processing

Figure 7: Architecture of unit selection TTS system. Copied from [7].

2.7 Applications of synthetic speech

Many applications could benefit from the use of synthetic speech. There have been products from talking toys and calculators to personal assistant applications in modern smart phones that could conduct a dialog with the user. In cases like warning and announcement systems, unrestricted vocabulary might not be required and simply recorded units of words and sentences could come to use. However for applications like reading applications for the blind, require a text-to-speech (TTS) system with unlimited vocabularies. [2]

Perhaps the most useful application of speech synthesis is to be used as a communication aid for the blind. Before the common use of speech synthesis, audio books used to be recorded by a professional speaker for use with the blind. This technique is expensive and time-consuming. Nowadays, there exist applications that could be used to read aloud information from a computer screen. [2]

Synthetic speech can provide voice for people who are born-deaf and people who have difficulties in speaking. Subsequently it provides a chance to communicate with people who do not comprehend sign language. Talking heads could improve the quality of communication since any visual type of information is very important for people who cannot hear. [2]

Another important area of application for synthetic speech is an educational situation. For example language learning applications could make use of synthesized speech to teach pronunciation. It could be used to teach people who have impairment for reading (dyslexics). Additionally synthetic speech could also be used together with spoken dialog systems in personal assistant applications, for proofreading with word processors, to read aloud email and SMS messages etc. [2]

3 Available tools and techniques

3.1 Available speech synthesis products and APIs

Many companies offer text-to-speech APIs to their customers in order to accelerate the development of new applications utilizing TTS technology. These companies include AT&T, IVONA, Neospeech and Readspeaker. In addition to these, major mobile operating systems like Android, IOS and Microsoft Windows offer API for text-to-speech.

A list of available speech synthesis products was found using an on line search. The following tools have been selected for testing, listed in alphabetical order. Sample voices from these applications have been attached with this document on a CD.

Acapela

Acapela Group is a company which develops text-to-speech software and services. It was formed from a combination of three companies that specialize in voice technology. Babel Technologies (Belgium), Infovox(Sweden) and Elan Speech (France). At the moment, Acapela has natural sounding voices for 25 languages including Finnish and has an API access for cloud-based TTS services. The service could be accessed though HTTP. Pricing is based on voice units. [11]

Espeak

Espeak is a compact open-source speech synthesizer that supports many languages. It is available for Linux and Windows. Since it uses a formant synthesis method, the synthesized voice is not natural sounding. Espeak is available as a command line utility or as a dynamic library (DLL). It has also been ported to other systems like Android and Mac OSX. [6]

Festival

Festival is a framework for building speech synthesis systems and offers full TTS through a number of APIs. It is distributed with an unrestricted license for commercial and non-commercial use alike. [12]

Google

The Google text-to-speech system is primarily developed for the Android operating system. It powers applications to read aloud the text on the screen and has support for multiple languages. Google services like Google Translate use the system. The service provides API to developers in the Android platform. There also exist an unofficial access to the API by using the Google Translate service. Google TTS has been implemented in the Chrome web browser to read any text in the browser.

Ispeech

ISpeech is a TTS service from a California-based company which develops speech solutions for various platforms. The cloud services are free to implement in mobile platforms while it costs a small fee to use a web platform. The service supports about 20 languages and the voice is of human-like quality. The company provides an API for developers for testing. [13]

Microsoft

Microsoft Speech API (SAPI) is an API developed by Microsoft and allows to use speech recognition and speech synthesis within Windows applications. The SDK is integrated into the Windows OS itself. The Microsoft Office and Microsoft speech server are among the applications that use SAPI. The Speech API is distributed freely and can be shipped with any Windows applications that use speech technology. The recent version of the API is SAPI 5. [14]

Nuance

Nuance is an American company that develops speech technology solutions. Its speech technology powers the Apple Siri personal assistant. Nuance speech technologies are available for embedded, mobile and cloud platforms. More than 40 languages are supported by the services. The on-line service for developers is accessed through HTTP REST interface. The developer API is available freely for evaluation purposes. [15]

ReadSpeaker

Readspeaker is a suite of web-based applications that use the TTS technology to enable speech on websites and mobile platforms. The application supports more than 35 languages. API for developers supports programming languages such as Java, Objective C, PHP, ASP and Flash. Free trial is available for evaluation. [16]

From the above speech synthesis systems, Festival and Espeak are open-source and freely available for any uses including commercial. The voice quality of these systems does not necessarily match with those available from commercial companies. However these systems allow their applications to be installed on one's own server without restriction. This implies that these could be deployed even without need to an Internet connection. The two systems have been selected for further use in the implementation phase.

Table 1: Text-to-speech products

Product	**Supported languages**
Acapela	more than 25
Espeak	many
Festival	more than 5
Google	many
Ispeech	over 20
Microsoft	many
Nuance	over 40
ReadSpeaker	more than 35

Text-to-speech products that support the Finnish language are summarized in Table 1.

3.2 Speech synthesis markup language

Speech Synthesis Markup Language (SSML) is an XML-based markup language for speech synthesis applications. To drive an interactive telephony service, it is often embedded in a VoiceXML scripts. However it may also be used alone for example in audio book creation. For assisting the generation of synthetic speech in web and other applications, SSML provides a rich markup language. The essential role of the markup language is to provide authors of synthesizable content with a standard way to control aspects of speech such as pronunciation, volume, pitch or/and rate across different synthesis-capable platforms. [17] SSML document processing is done in six steps. These are shown in Table 2.

Table 2: Steps in SSML document processing

No.	Step	Description
1	XML Parse	extracting document tree from content
2	Structure analysis	formatting the way document should be read
3	Text normalization	converting written to spoken form
4	Text to phoneme	words to the smallest unit of sound
5	Prosody analysis	set of features of speech such as pitch and speaking rate
6	Waveform production	audio waveform generation

The following listing shows a sample SSML.

```
<speak xml:lang="en-US">
<paragraph>this is a test</paragraph>
<paragraph xml:lang="fi">tämä on testi</paragraph>
</speak>
```

Listing 1: Sample SSML syntax

A text-to-speech system that supports SSML is responsible for rendering a document to the spoken output and for using the information in the markup to render the document as intended by the author. SSML document creation could be done automatically, by a human author or both.

3.3 Speech quality and evaluation

Intelligibility, naturalness and suitability can be used to evaluate the quality of synthetic speech. In some applications like reading used for the blind, intelligibility at a high rate of speed is more important. On the other hand in multimedia applications, naturalness and prosodic features are more important. Evaluation could be carried out at various levels such as the phoneme, word or sentence level based on what kind of information is required. [2]

Text processing and linguistic realization are important in a text-to-speech system just like the acoustic properties. For a good result different kinds of techniques should be used. Evaluation is often made by a subjective listening test with a response set of syllables, words and sentences. Usually consonants are more problematic to synthesize than vowels. Hence most test material is targeted to test consonants. Especially nasalized consonants such as (/m/ /n/ /ng/) are usually those which result in problems. When a test procedure is repeated for the same listening group, the test results may increase significantly. This is because after every listening session the testers get more familiar with the synthesized speech and understand it better. [2]

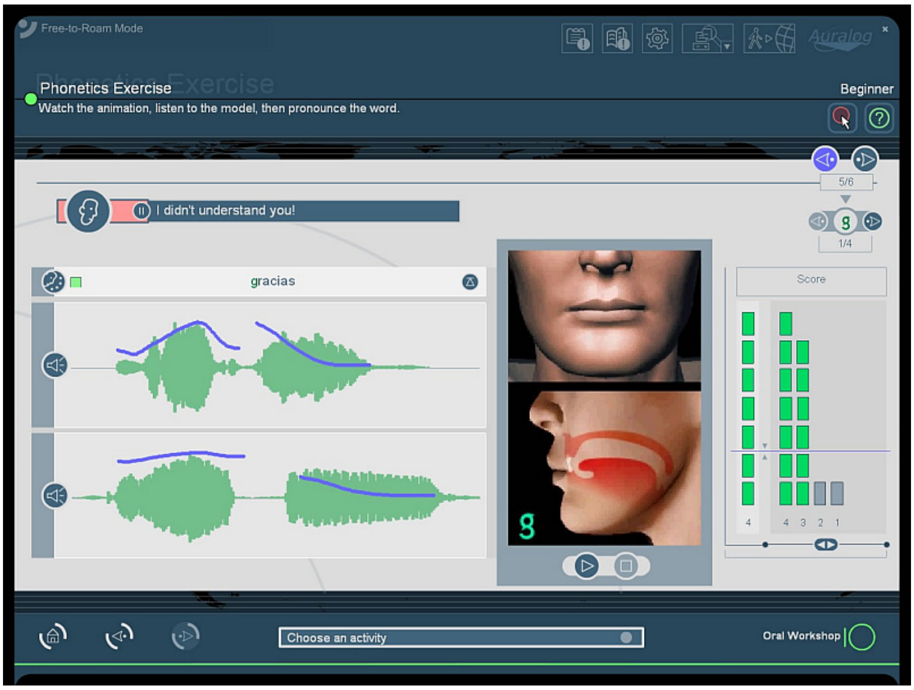
4 Current research in speech technologies

The future of speech technologies depends on the current research undertaken in the field and identifying the promising directions. Since the 1990s advancements in computer software and hardware along with a better understanding of speech has allowed speech technologies to be deployed in enterprises. At the same time developments in the field of linguistics have led the use of computers in the use of developing listening and speaking skills and second language acquisition. Some major language learning applications these days are using these technologies and new standards are being developed in this area. [18]

4.1 Visualization of speech

There have been tools to provide graphical representation of speech in computers since the 1990s. These were intended for users to compare their pronunciation along with a native speaker of a language. However a little guidance was left to the user on how to improve pronunciation. Most language learning systems follow a similar model in that learners first listen to a native speaker and are then asked to produce their own speech. [18]

Waveform representation of speech is one way to provide feedback to the user. In addition to this the feedback could also be audiovisual including representation of mouth parts. One of the issues in traditional approaches to visualization of speech is the difficulty users may have in understanding or interpreting the feedback displays. Some speech software developers have experimented with game-like interfaces in which a user controls a game character based on his/her correct pronunciation. In some ways a computer-based pronunciation practice is more efficient. These include, when there is one-to-one access for students with a computer, when users are allowed to proceed with their own pace and when a computer program is customized to the needs of the student. [18]



Phonetics Exercise

Watch the animation, listen to the model, then pronounce the word.

I didn't understand you!

Figure 8: Language learning application implementing visualization of speech. Copied from [19].

Figure 8 shows a program known as "Tell me more". This program implements speech recognition in order to provide a score based on how close the user pronunciation is relative to a native speaker's recording. In the program visual feedback is provided using audio waveforms and showing mouth parts used in the speech visualization.

4.2 Automatic speech recognition (ASR)

In the past decades, ASR has been an area of considerable interest in the signal processing and human language technology (HLT) communities. Despite the growing practical applications a significant development in ASR is expected in the future. It is expected to be worldwide deployed. This could be expected due to the growth of technology in infrastructure, knowledge representation and increasingly advancing algorithms. Moore's law predicts that the amount of computations achievable for given cost is doubled in every 12 to 18 months as well as the cost of memory shrinks. These developments have enabled researchers to run more complex algorithms in a short time. Increasing the availability of speech corpora has led automated systems to achieve proficiency. For instance, various national institutions along with educational ones have provided research tools such as Hidden Markov Model toolkit (HTK) and Sphinix. [20]

Audio indexing and mining have enabled high-performance automatic topic detection as well as applications for automatic speaker detection. Current ASR systems perform poorly when encountering audio signals that differ from the limited conditions under which they were developed. This focused research area would concentrate on creating and developing systems that would be more robust against variability and shifts in acoustic environments, such as environmental noise and various speaker conditions. [20]

A major change in speech analysis was the introduction of statistical methods like Hidden Markov Model (HMM), 30 years ago and still prevails. A number of new models have been introduced and expected in the future to come. Rapid portability to emerging languages is an other domain. Today's top ASR systems are not readily available for all languages and hence the goal in the future is in the direction that spoken language technologies are easily portable. [20]

4.3 Standards

Integrating a speech technology to a software has been difficult due to the fact that there have not been commonly accepted standards. However this has been changing recently because of efforts made by various sectors. W3C's (World Wide Web Consortium) effort to develop speech-related standards can be a good example. Some interesting projects that use the web include SPICE from CMU whose main merit is to create an ASR for rare languages. These projects use the web and they are able to learn and improve through time. Increased use of multimedia such as videos together with speech technologies and incorporation of natural speech is becoming common. Virtual reality games have been increasingly using speech technologies and an increasing implementation of speech technologies in mobile technologies is expected to continue. [18]

Table 3: Some institutions conducting research in speech technologies

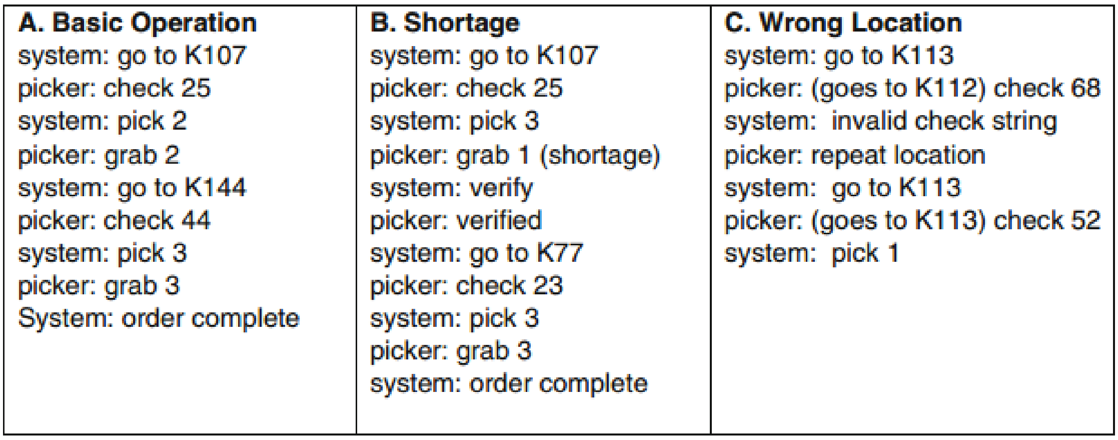
Institution	**URL**
IBM research,	http://www.research.ibm.com/tts/
Cambridge	http://mi.eng.cam.ac.uk/research/dialogue/
Microsoft research	http://research.microsoft.com/en-us/projects/whistler
John Hopkin university	http://www.clsp.jhu.edu/
AT&T Natural Voices	http://www2.research.att.com/ ttsweb/'/demo.php#top

Some of the institutions which are conducting research in speech technologies are shown in Table 3.

4.4 Emerging markets in mobile speech

There is a growing diverse market for mobile speech technologies. Mainly these new markets include Warehouse operations, offender monitoring and robotics. With the emergence of Siri, recent focus has been on personal assistants for smart phones. This has led the creation of competitors to Siri such as Nina (Nuance communication), Lola (SRI international), Lexee (Angel labs) and Watson for smart phones(IBM). [21]

"Eyes busy", "hands busy" environments in factories and warehouses were among the earliest markets for mobile speech. These include manufacturing inspection, sorting, order picking, return processing etc. A workers eyes and hands need to be focused on the tasks to safely perform the tasks so that errors and accidents do not occur.
Starting from the 1980s these eyes-busy tasks were well suited to speech technologies. The alternatives like pausing tasks to write findings on data-sheet is error prone or even using additional data-clerks to write using for example a laptop is costly. Since factory users do repetitive tasks every day the vocabularies for speech technologies could be limited and users could train their own voices. For example figure 9 on the next page provides an example of spoken dialog interaction of a factory worker with a system. In these factory environments the main challenge has been noise. For this, speech technology companies have for example used noise canceling microphones and sometimes embedded the mic into hard-hats or protective ear-ware.[21]



A. Basic Operation	B. Shortage	C. Wrong Location
system: go to K107	system: go to K107	system: go to K113
picker: check 25	picker: check 25	picker: (goes to K112) check 68
system: pick 2	system: pick 3	system: invalid check string
picker: grab 2	picker: grab 1 (shortage)	picker: repeat location
system: go to K144	system: verify	system: go to K113
picker: check 44	picker: verified	picker: (goes to K113) check 52
system: pick 3	system: go to K77	system: pick 1
picker: grab 3	picker: check 23	
System: order complete	system: pick 3	
	picker: grab 3	
	system: order complete	

Figure 9: Order picking. Example drived from demo video produced by Voxware (copied from [21])

Correction facilities ware among the largest market for speaker verification. The main reason for using this technology is the increase in alternative sentencing for non violent offenders. Heavy caseloads made it increasingly difficult for officers to monitor offenders effectively.[21]

Despite the extreme capabilities of robots in speech in various science fiction, there exist few actual robots that utilize speech. There exist two categories: robots which are able to respond to pre-defined speech and those who can learn and use language (autonomous robots). There exist toys which can accept commands via voice and respond either in action or using speech synthesis. Additionally there are worker robots developed for special needs people. For example hospital bed controlled by voice. [21]

5 Implementation

Many speech technology companies provide TTS solutions. From these there exist a number of natural sounding Finnish text-to-speech systems. Designing a new text-to-speech system from scratch is beyond the scope of this thesis. However some existing text-to-speech systems will be used in an experement to build a web based application intended in assisting language learning. The main features of this application will be synthesizing user text into speech and using speech recognition for pronunciation practice. Text-to-speech engines will be installed on a server and the client side app communicates with the server using HTTP GET method. Overall the implementation plan is such that a user entered text is sent to the server first. Then after a speech is synthesized on the server the resulting audio is sent back to the client side. Figure 10 illustrates the client and server architecture.

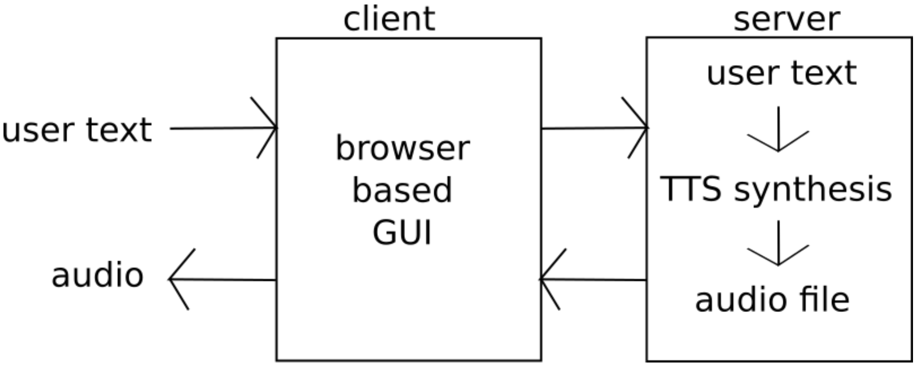


Figure 10: Client and server architecture

5.1 Setting up a Linux text-to-speech server

In order to have a web interface to the speech engines installed, a server environment is set up. The server for this demo was Ubuntu 14.04 LTS and it was installed as a guest system into a virtual machine. A complete LAMP (Linux Apache MySQL and PHP) stack was installed in the server environment. On the server side PHP script accepts user text from browser and returns synthesized audio. On the other hand Jquery and Javascript were used for the client side web application.

Espeak and Festival text-to-speech systems were selected for the experement. Espeak supports various languages including Finnish. It can be easily installed to a Linux system from Linux repositories. Similarly Festival is found in the Ubuntu package repositories. However Finnish voices for festival are not pre-installed along with festival and have to be downloaded separately. Finnish Festival voices have been developed in the University of Helsinki as part of Suopuhe project and are available in Ubuntu and Debian repositories.

5.1.1 Espeak

Espeak is an open source software and is available for Linux, Windows and MacOSX operating systems. Additionally there exists an Espeak port to the android platform.

Espeak is a formant type rule based speech synthesizer. A text to be synthesized is first converted into phoneme. Phoneme is the smallest linguistic unit. There exist different set of phonemes for different languages. For example English has 41 phonemes as defined by International Phonetic Association (IPA). Espeak uses some rules to convert a text to the phonemes and allophones (variation of phoneme). Hence the program has smaller memory consumption when compared with dictionary based speech synthesis systems.

Below is a list of some command-line parameters of Espeak. For a simple text synthesis the following is used

espeak "this is a test" or **espeak** -f <text file>

The basic syntax is as follows

espeak [options] ["text words"]

Some basic options for synthesis include:

-w <wave file>

writes the speech output to a file in a WAV format instead of speaking it aloud.

-v <voice filename>[+<variant>]

sets voice for the speech, usually to select a language. For example the following command sets a Finnish voice.

espeak -vfi

In addition to these female and male voices are synthesized using variable pitch. These include +m1 +m2 . . . +m7 for male voices and +f1 . . . +f4 for female voices.

Additionally Espeak has parameters to control the synthesized voice's speed, pitch and amplitude.

Espeak allows users to add or improve additional languages. This consists of defining phoneme table for a language, spelling to phoneme translation rules and adding a file for pronunciation of symbols, numbers and abbreviations. Espeakedit is a program used prepare phonemedata and compile voices for espeak. A screenshot of espeakedit is shown in figure 11 on the next page

5.1.2 Festival

Festival is a multilingual text-to-speech system developed originally by Alan W. Black at University of Edinburgh. Festival works in two modes. Command mode and text-to-speech mode. When in command mode input (from file or interactively) is interpreted by the command interpreter. When in text-to-speech mode, input is treated as text to be rendered as speech. Festival's basic calling method is like:

festival [options] file1 file2 ....

In command driven mode festival is started with no arguments. some useful command line options of festival are the following:

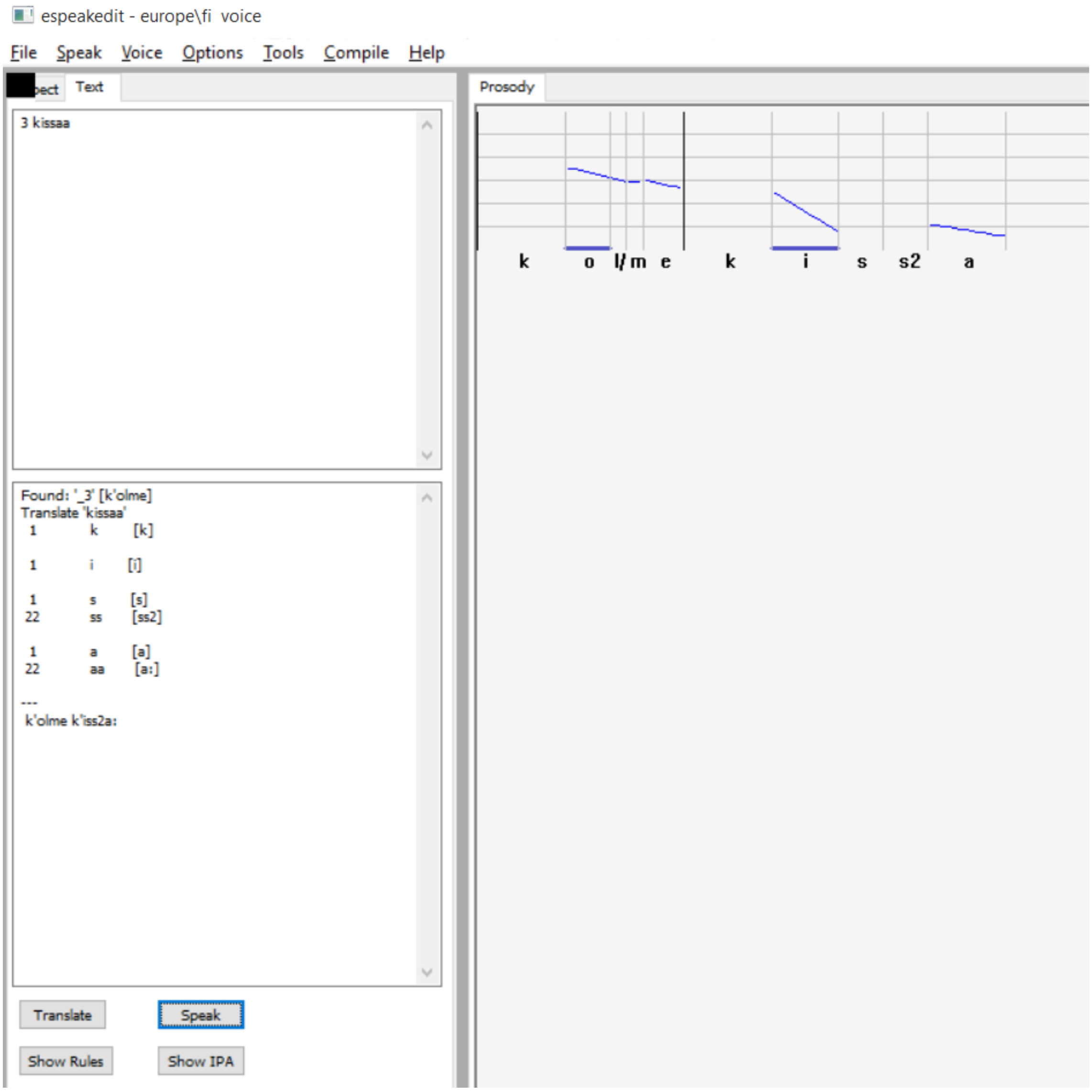
--tts

synthesize text in files as speech, if no file is given from standard input (stdin)

--language <string>

Run in named language

After eSpeak and Festival have been installed on the server, a small demo web application that uses these two engines is implemented.



```
3 kissaa
```

```
Found: '_3' [k'olme]
Translate 'kissaa'
 1        k      [k]

 1        i     [i]

 1        s     [s]
22        ss    [ss2]

 1        a     [a]
22        aa    [a:]

---
 k'olme k'iss2a:
```

Figure 11: Espeakedit window

5.1.3 Server side scripts

On the server side PHP scripts are used to call the command line parameters of Espeak and Festival. These scripts accept a text as a GET parameter and pass it to either Espeak or Festival command line applications. Then the scripts return an audio from the synthesized voice in MP3 format. Sample server side code is shown in the listing below. Note here that **text2wave** comes along with festival installation and converts text to an audio in WAV format.

```php
<?php
$base_dir = '/tmp/';
$text = $_GET['text'];
$filename = md5($text) . '.mp3';
$filepath = $base_dir . $filename;
$text = escapeshellarg($text);
if (!file_exists($filepath)) {
  $cmd = "echo $text| iconv -f UTF-8 -t ISO8859-1 | text2wave |\
  lame --preset voice -q 9 --vbr-new - $filepath";
  exec($cmd);
}
header('Content-Type: audio/mpeg');
header('Content-Length: ' . filesize($filepath));
readfile($filepath);
?>
```

Listing 2: PHP script to use commandline festival

5.2 Developing demo web application

The client side application has three main pages. The menu page has links to these pages. These are "Read Stories" and "Listen Dialogue" and "Practice pronunciation". The initial menu page is shown in Figure 12.

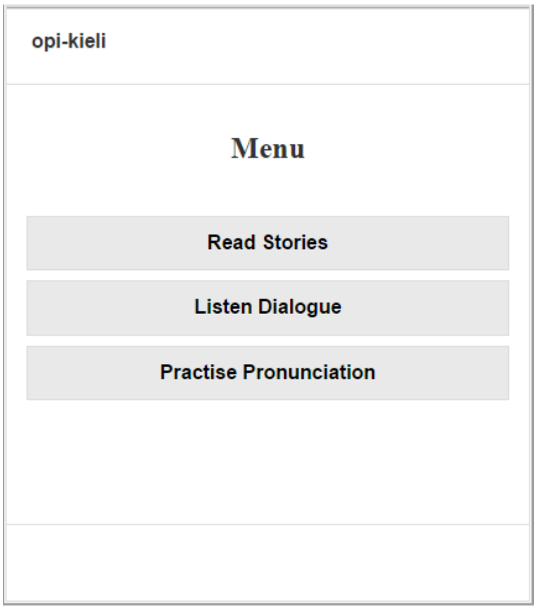


Figure 12: Applications menu page

Read stories page is a place where user can hear to stories read aloud by text-to-speech engines. On this page a user could input his/her own text to the blank space and then listen a story. Sample text could also be fetched by the user for testing. Figure 13 shows the Read stories page.

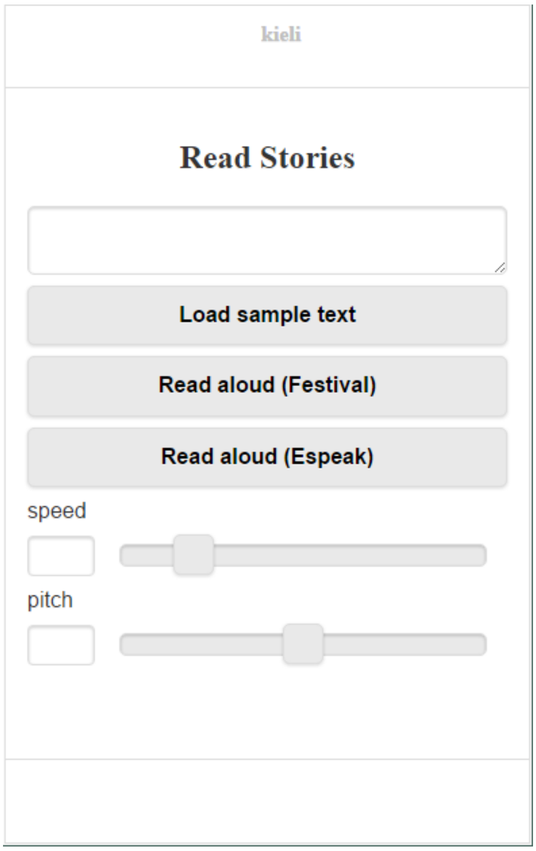


Figure 13: Read Stories page

The load stories page leads to a list of stories. These stories are saved as text files on the server and loaded with using Ajax synchronously.
Below is a function that loads text file into a browser.

```
function loadfile(filename) {
    /*
    Function: loadfile
    loads file given on the parameter using ajax synchronous
    Parameters:
    filename - the file name to be loaded
    See Also:
    */
    $.ajax({
        async : false,
        url : filename,
        success : function(data) {
            story1 = data;
        }
    });
}
```

Listing 3: Function that text file synchronously using ajax

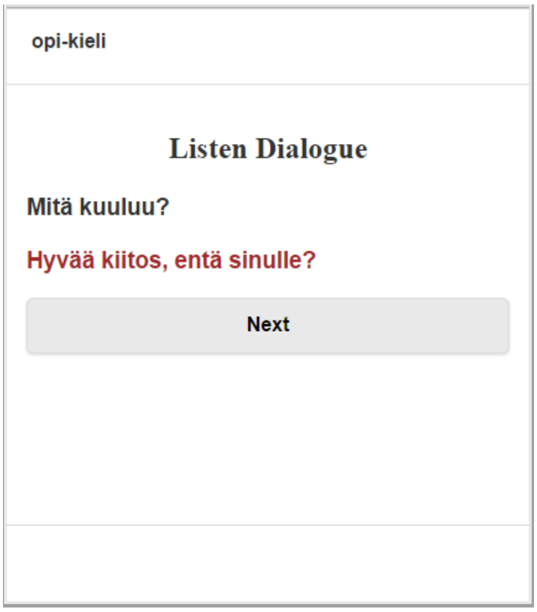
On the client side the following javascript function sends text to be synthesized and plays the resulting audio obtained from the server.

```
function speak(text) {
    /*
    Function: speak
    speaks aloud using an on-line speech service.
    See Also:
    */
    var audio_src = "http://ttss.com/f.php?text='"+text+"'";
    var audio = document.getElementById('speakaudio');
    audio.src = audio_src;
    audio.play();
}
```

Listing 4: Function that sends text to server

The Listen Dialogs has conversational phrases between two speakers. A set of dialogs is shown on a given page. User clicks on the dialogs to hear it aloud. A sample dialogs page is shown in figure 14 on the following page.

The practice pronunciation page features speech recognition that uses Google's technol-



opi-kieli

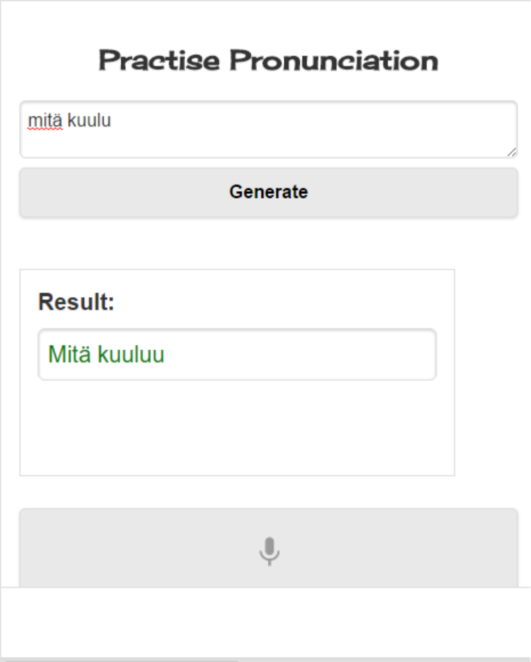
Listen Dialogue

Mitä kuuluu?

Hyvää kiitos, entä sinulle?

Figure 14: Listen Dialogue page

ogy as a back end. On this page a user could try a word with a speech synthesizer and at the same time practice pronunciation using a microphone as an input.



Practise Pronunciation

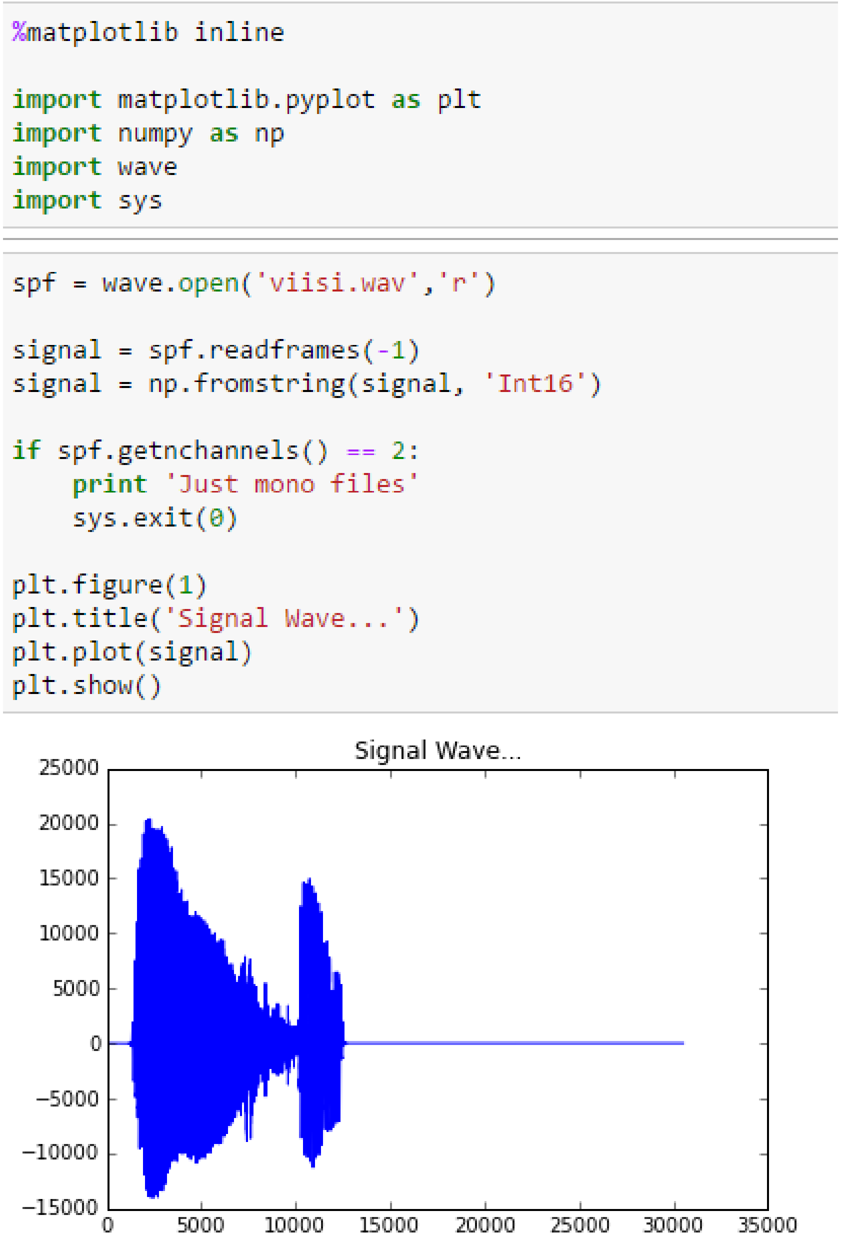
mitä kuulu

Result:

Mitä kuuluu

Figure 15: Practice Pronunciation page

On the client side, Jquery mobile was used to design the demo web application. Jquery mobile allows creation of responsive web based mobile applications.



```
%matplotlib inline

import matplotlib.pyplot as plt
import numpy as np
import wave
import sys
```

```
spf = wave.open('viisi.wav','r')

signal = spf.readframes(-1)
signal = np.fromstring(signal, 'Int16')

if spf.getnchannels() == 2:
    print 'Just mono files'
    sys.exit(0)

plt.figure(1)
plt.title('Signal Wave...')
plt.plot(signal)
plt.show()
```

Figure 16: Python script to generate audio waveform

Generating a waveform image from a user's audio input was one of the initial ideas in the project. Python based script was used to generate a waveform from audio recording. Here a user could produce visuals from synthesized audio to compare it with his own speech. Despite the fact that this solution was not integrated into the project, it was found important during development for demo purposes. Figure 16 shows a script that generates audio waveform from an audio input.

5.3 What was achieved

At the moment speech technologies are developing in a fast pace, specially in the mobile market. In this project historical background, available techniques, current research and application areas of speech synthesis technology have been investigated.

There are quite many speech synthesis engines available on the market. Some products have been selected and compared. Then two of them have been selected for further experiment. The main reason in the selection has been ease of availability for off-line use, being free of cost and open source.

Speech technologies have an important role in language learning. A small web application has been developed in order to demonstrate the application of speech synthesis for education. Hence in summary two text-to-speech applications, Espeak and Festival have been selected for use in a small web application intended in assisting language learning.

6 Conclusions

At the moment there exist many speech technology companies offering products for the market. Additionally there is a lot of research in companies and educational institutes in the field. As a result the quality of speech synthesis systems has improved a lot that many systems could produce natural sounding and very intelligible voice. Some companies have even produced TTS solutions with children's voice. However some issues still exist like producing the right emotions in the synthesized speech. On the other hand several new applications that incorporate speech technologies have been developed. Apple's SIRI and Google Now have brought speech technologies for the mass and this has gained popularity as an alternative interaction method with mobile devices. Other new emerging markets in the industry include offender monitoring, robotics and warehouse operations.

Basic methods of speech synthesis have been covered in chapter two. These include formant based, concatenative and articulatory synthesis. Concatenative method has gained more popularity because more natural sounding output could be produced. On the other hand formant based synthesis is known for its smaller memory footprint and flexibility. Hence it has been popular in embedded technologies. Articulatory synthesis technology is still in research phase due to its complexity.

In this project a demo web application has been developed to illustrate the use of speech technologies in language education. This demo features two text-to-speech applications; Espeak and Festival. A user is able to use the application to read text aloud. It has been found that TTS technologies are important in such applications designed to assist language education.

There exist many natural sounding text-to-speech technologies for many of the European languages. However speech technologies are not equally available in every language. Some experiments were carried out to use Oromo language, one of the languages in

Ethiopia, to read a text using a Finnish speech synthesis technology. From the experiments the output sound was close enough to good. This may have been due to similar features in vowel length and consonant gemination between the languages.

Many of modern speech technology products are complex that it is difficult task to port them among languages. This is specially true when the speech technology is based on a recording, such as unit selection speech synthesis system. Hence portability of speech technologies among languages remains a work for the future.

References

1 Measuring and modelling speech; 1998. [Online; accessed 21-01-2016]. Available from: http://www.haskins.yale.edu/featured/heads/mmsp/intro.html.

2 Lemmetty S. Review of speech synthesis technology. Helsinki University of Technology. Espoo; 1999. Available from: http://research.spa.aalto.fi/publications/theses/lemmetty_mst/thesis.pdf.

3 Rose D, Dalton B. Plato revisited: Learning through listening in the digital world. Recording for the Blind & Dyslexic. 2007;[Online; accessed 21-01-2016]. Available from: http://www.udlcenter.org/sites/udlcenter.org/files/Plato_Revisited.pdf.

4 Allen J, Hunnicutt MS, Klatt DH, Armstrong RC, Pisoni DB. From text to speech: The MITalk system. Cambridge University Press; 1987.

5 Burk P. Music and Computers: A Theoretical and Historical Approach: Course Guide. Key College Pub.; 2005. [Online; accessed 21-01-2016]. Available from: http://music.columbia.edu/cmc/MusicAndComputers/.

6 Duddington J. eSpeak Text to Speech; 2008. [Online; accessed 21-01-2016]. Available from: http://espeak.sourceforge.net.

7 Chalamandaris A, Karabetsos S, Tsiakoulis P, Raptis S. A unit selection text-to-speech synthesis system optimized for use with screen readers. Consumer Electronics, IEEE Transactions on. 2010;56(3):1890--1897.

8 Silen H, Helander E, Koppinen K, Gabbouj M. Building a Finnish unit selection TTS system. In: SSW; 2007. p. 310--315.

9 Bachan J. Efficient Diphone Database Creation for MBROLA, a Multilingual Speech Synthesiser. Institute of Linguistics, Adam Mickiewicz University, XII International PhD Workshop. 2010;Available from: http://mechatronika.polsl.pl/owd/pdf2010/303.pdf.

10 Hubbard P. Computer Assisted Language Learning: Critical Concepts in Linguistics. Present Trends and Future Directions in CALL. Routledge; 2009.

11 Acapela FAQ; 2015-10-09. [Online; accessed 21-01-2016]. Available from: http://www.acapela-group.com/voices/faq/.

12 The Festival Speech Synthesis System;. [Online; accessed 21-01-2016]. Available from: http://www.cstr.ed.ac.uk/projects/festival/.

13 iSpeech API; 2013. [Online; accessed 21-01-2016]. Available from: http://www.ispeech.org/api.

14 Microsoft Speech Platform;. [Online; accessed 21-01-2016]. Available from: https://msdn.microsoft.com/en-us/library/jj127858.aspx.

15 Nuance, Text to speech;. [Online; accessed 21-01-2016]. Available from: http://research.nuance.com/category/text-to-speech/.

16 ReadSpeaker, About Us;. [Online; accessed 21-01-2016]. Available from: http://www.readspeaker.com/about-us/.

17 Mark R Walker AH. Speech Synthesis Markup Language; 2001. [Online; accessed 21-01-2016]. Available from: http://w3.org/TR/2001/WD-speech-synthesis-20010103/.

18 Godwin-Jones R. Emerging technologies: Speech tools and technologies. Language Learning & Technology. 2009;13(3):4--11.

19 Tell Me More Review;. [Online; accessed 21-01-2016]. Available from: http://www.effectivelanguagelearning.com/language-course-reviews/tell-me-more-review.

20 Baker J, Deng L, Glass J, Khudanpur S, Lee CH, Morgan N, et al. Developments and directions in speech recognition and understanding, Part 1 [DSP Education]. Signal Processing Magazine, IEEE. 2009 May;26(3):75--80.

21 Markowitz J. Beyond SIRI: Exploring Spoken Language in Warehouse Operations, Offender Monitoring and Robotics. In: Neustein A, Markowitz JA, editors. Mobile Speech and Advanced Natural Language Solutions. Springer New York; 2013. p. 3--21. Available from: http://dx.doi.org/10.1007/978-1-4614-6018-3_1.

1 List of websites for TTS companies

Table 4: text-to-speech products

Product	URL
Acapela	http://www.acapela-group.com/
Espeak	http://espeak.sourceforge.net/
Festival	http://www.cstr.ed.ac.uk/projects/festival/
Google	https://www.google.com/intl/en/chrome/demos/speech.html
Ispeech	http://www.ispeech.org/
Microsoft	http://www.microsoft.com/en-us/download/details.aspx?id=27225
Nuance	http://www.nuance.com/index.htm
ReadSpeaker	http://www.readspeaker.com/
Svox	http://www.nuance.com/products/SVOX/index.htm