Jolanta Bachan

Specjalność: Językoznawstwo i Informacja naukowa

Close Copy Speech Synthesis for Perception Testing and Annotation Validation

Praca magisterska napisana pod kierunkiem
profesor doktor habilitowanej inżynier
Grażyny Demenko

Uniwersytet im. Adama Mickiewicza w
Poznaniu Wydział Neofilologii
Instytut Językoznawstwa

Poznań 2007

Acknowledgements

I wholeheartedly thank Professor Grażyna Demenko and Professor Dafydd Gibbon for their help, their training and assigning me challenging tasks to fulfil. Without their supervision carrying out this research would not have been possible.

I would like to thank Professor Piotra Łobacz for the invaluable knowledge which she passed on me during her phonetics class and to Professor Andrzej Pluciński for his statistical advice.

Thanks also to Dr. Katarzyna Klessa-Francuzik and Agnieszka Wagner M.A. for their help and instructions on annotations and to Małgorzata Mazur for her cooperation on the speech perception tests for children with a cochlear implant.

Further thanks to staff and students at Bielefeld University who took very good care of me during my stay there as an Erasmus student. Special thanks to Dr. Thorsten Trippel and Arne Hellmich for their help, advice and constructive suggestions.

I would like to thank all the people who took part in my research: children from the preschool "Wesoły domek", children with the cochlear implants, children with the hearing aids, my family and friends.

And finally, immeasurable thanks to my parents and my brother for their unfailing support and for the part they have played in my education.

Bielefeld, 21.05.2007

OŚWIADCZENIE

Ja, niżej podpisana

Jolanta Bachan

studentka Wydziału Neofilologii

Uniwersytetu im. Adama Mickiewicza w Poznaniu

oświadczam,

że przedkładaną pracę dyplomową

pt: Close Copy Speech Synthesis for Perception Testing and Annotation Validation

napisałam samodzielnie.

Oznacza to, że przy pisaniu pracy, poza niezbędnymi konsultacjami, nie korzystałam z pomocy innych osób, a w szczególności nie zlecałam opracowania rozprawy lub jej istotnych części innym osobom, ani nie odpisywałam tej rozprawy lub jej części od innych osób.

Oświadczam również, że egzemplarz pracy dyplomowej w formie wydruku komputerowego jest zgodny z egzemplarzem pracy dyplomowej w formie elektronicznej.

Jednocześnie przyjmuję do wiadomości, że gdyby powyższe oświadczenie okazało się nieprawdziwe, decyzja o wydaniu mi dyplomu zostanie cofnięta.

Jolanta Bachan

Table of Contents

CHAPTER 1 Introduction

1.1 Objectives

The aim of the present study is, first, to develop a restricted domain speech synthesis concept for automatically generating acoustic stimuli for future use in evaluating cochlear implants for children, second, to implement a prototype synthesiser, and third, to validate the quality of the annotated speech corpus on which the system will be based. The main motivation for including a speech synthesiser in the system is to increase the flexibility of the available test stimuli.

In the present form of the evaluation tests of cochlear implants recorded stimuli are used. But a recorded custom corpus of recordings is static and inflexible. The solution is to use speech synthesis which is dynamic and flexible. Moreover, synthetic speech allows presentation of carefully controlled speech-like stimuli to listeners in order to obtain judgements on their speech perception (Ashby & Maidment 2005: 181). The approach taken is to use *Close Copy Speech* (CCS) synthesis whose functionality is very useful and practical:

1. CCS synthesis gives a basis for high quality Text-To-Speech (TTS) synthesis with prosodic parameter manipulation.
2. CCS synthesis is used for validation of annotations:
    1. software assessment – testing annotations created by an automatic annotator,
    2. speech output assessment – testing quality of synthesised speech against the gold standard of synthetic speech which is provided by the MCCS and ACCS synthesis procedures. MCCS and ACCS synthesis methods create

a best case bench mark to which synthetic speech generated by speech synthesisers of different kinds are referred.

The basis for the synthesiser is the *Close Copy Speech* (CCS) synthesis or resynthesis method, in which it is the task of the synthesiser to "repeat utterances produced by a human speaker with a synthetic voice, while keeping the original prosody" (Dutoit, 1997: 134). In this method, "close copy" means that the synthetic speech is as similar as possible to a human utterance.

In fact, in the present context, "copy" means that the input to the synthesis engine for a given utterance is derived directly from a corresponding utterance in the annotated corpus data. The method is taken one step further, in the present approach, in parametrising the prosody, so that modifications of the original prosody (speech timing and pitch patterns) can also be systematically introduced in future work.

In the present study, the definition by Dutoit is interpreted to mean that the Natural Language Processing or Text-To-Speech (TTS) component of the synthesiser is replaced by an analysis of a recorded speech signal. The analysis in this context consists of a recorded speech signal, a method for pitch extraction from the speech signal, and a time-aligned phonemic annotation of the speech signal. The development procedure used in the present study has three phases:

1. Manual Close Copy Speech (MCCS) synthesis: manual transfer of parameters from the original signals and annotations to the synthesiser interface.
2. Automatic Close Copy Speech (ACCS) synthesis: automatic transfer of parameters from the original signals and annotations, based on specifications derived from the MCCS phase.
3. Parametric Close Copy Speech (PCCS) synthesis: interactive and automatic parametrisation at the ACCS derived synthesiser interface.

The present thesis reports on the background to the development, and on the MCCS and ACCS phases of the development. As far as PCCS synthesis is concerned, one parametric manipulation has been used in this research, namely manipulation of the pitch patterns. The manipulation of the durations have also been investigated, but are not of concern of the present thesis.

For the purposes of this study, MBROLA, a *de facto* standard diphone synthesis

engine with a suitably modular language-to-speech interface, was selected (Dutoit et al. 1996: 1393-1396). The MBROLA software was chosen because:

1. a Polish diphone database is available (PL1 - A Polish female voice for the MBROLA synthesiser),
2. a clear language-to-speech interface (PHO file) is used,
3. MBROLA allows easy manipulation of duration and prosody parameters.

1.2 Overview

The thesis begins with the description of the use cases which feed into the CCS synthesiser development. The arguments for the introduction of speech synthesis in the speech perception tests are outlined, but the actual speech synthesis component has not been implemented into the existing set of speech perception tests. This task is beyond the scope of the present research.

The next chapter includes an overview of Text-To-Speech (TTS) systems and methods. In Chapter 4 requirements for CCS synthesis are presented. The design of CCS synthesis is outlined in Chapter 5, together with presentation of the MBROLA speech synthesiser. Chapters 6 and 7 describe the architectures and implementation of MCCS and ACCS synthesis respectively.

Chapter 8 deals with speech output evaluation methods. A list of criteria for the TTS evaluation and a subset of these criteria for CCS evaluation purposes are outlined. Moreover, diagnostic evaluation of the developed program for ACCS synthesis system is presented and tests of naturalness and intelligibility of speech, together with their results are included. Simultaneously, the quality of the annotated speech data which is used for CCS synthesis is tested.

The thesis ends with conclusions drawn from this study and future stages of the CCS development and implementation.

CHAPTER 2 Context for Close Copy Speech synthesis development

2.1 Context of the TTS development

The context of the present development is a project for testing the functionality of cochlear implants in children. The project strategy involves the development of tests supported by software, administration of the tests to normal children, children with hearing aids, and to children with cochlear implants. An overview of the context is shown in Figure 1; the individual components are needed for defining the use cases and the use case based requirements for the speech synthesiser.

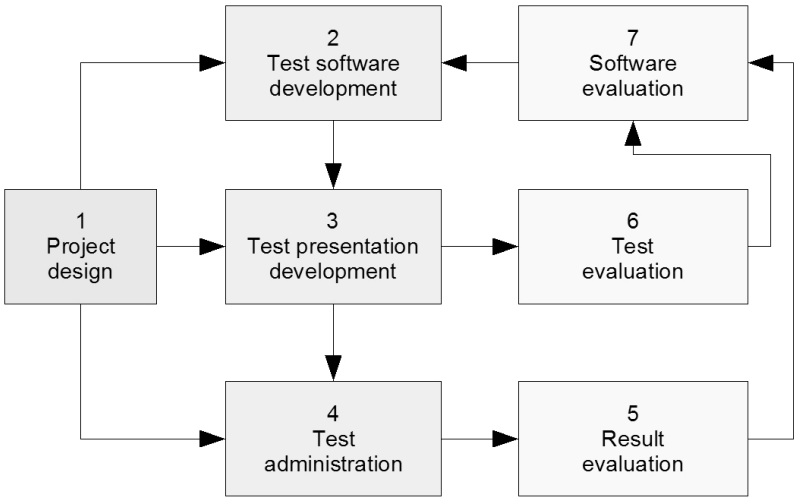

Figure 1: Project context for TTS software development.

2.2 Overview

The present study is concerned with developing a Close Copy Speech synthesis subcomponent for component #2 shown in Figure 1. Evaluation feedback is expected

from all other components. The components #3, #4, #5, #6 and #7 serve to define use cases for deployment of the TTS software; the main use cases considered are #3, test presentation development and #4, test administration.

2.3 Requirements: use cases

In the present section use cases of the speech perception tests for children with a cochlear implant are described. The descriptions show the need for implementing the speech synthesis component to the tests, which would make the test more precise, attractive and flexible.

2.3.1 Use case: Test presentation development (component #3)

The battery of speech perception tests for children with a cochlear implant was created at Adam Mickiewicz University. In the project, linguists, phoneticians, graphics designers and computer programmers were involved. The tests were designed in close cooperation with experts from the Medical Academy, and audiologists from the Marke-med centre, both in Poznan. The tools for administering these tests contain two types of speech perception tests:

1. *Nonsense stimuli tests*: tests with nonsense stimuli. Some of the tests in this set make use of synthesised stimuli. The aim of these tests is to assess whether the subject is able to take the verbal tests.
2. *Verbal stimuli tests*: tests with verbal stimuli. The tests make use of recorded stimuli produced by different human speakers. The tests are composed of words and sentences uttered in isolation.

Both sets of tests examine children's perceptive and linguistic skills making use of acoustic signals only. There are no visual cues in the test procedures, so the subject cannot lip-read. In both kinds of test the subject answers by pointing at a picture on a computer screen. The tests were designed for young children and touch screens were provided, so that children who did not know how to use a computer mouse could also take the tests.

The tests with verbal stimuli are designed for children who are able to comprehend speech, but who may be unable to give verbal responses. In these tests six different voices were used to test intelligibility of different voice pitches. The

tests make use of the following voices: two male adults, two female adults, one male child, one female child. The tests are listed in Appendix A. The description of the most important tests from the set of speech perception tests for children with a cochlear implant is provided below (cf. Ogórkiewicz et al. 2005, Bachan 2006):

Discrimination and identification of quantity. The aim of this test is to discriminate and identify the quantity of vowels. During the test synthetic syllable ***be*** is generated with different acoustic parameters simulating male, female and children's voice, the vowel ***e*** has two different lengths: the short ***e*** (50ms), and the long ***eee*** (150ms). Pictures of a short and a long sheep are used imaging the duration of the vowels.

Identification of disyllabic words of structures: *cvcv* (np. woda), ***cvccv*** (np. łóżko), ***cvcvc*** (np. banan), ***ccvcv*** (np. klucze), ***ccvcvc*** (np. sweter). It this test resynthesised stimuli are used. All the stimuli have a flat fundamental frequency contour.

Identification of unstressed syllables. The aim of this test is to assess the ability of identifying mono- and disyllabic words pronounced separately differing in the unstressed syllable only. E.g. *król – król**ik***, *kran – **e**kran*.

Identification of voice. The aim of this test is to recognise speaker's voice. The male, female and children's voice is used. The child responds by pointing out the proper picture of a man, a woman or a child.

Identification of segmental characteristics – vowels and consonants. The aim of this test is to differentiate and to identify individual vowels or individual consonants in different contexts of a word. The test material is composed of minimal pairs - pairs of words that differ in one phoneme only. E.g. for vowels: *m**a**ska - m**i**ska*, *b**a**t – b**u**t*, for consonants: *ła**p**a – ła**t**a*, ***b**eczka – **t**eczka*.

Identification of segmental characteristics in words and logatoms. The aim of this test is to examine the perception of segments of speech. The test material is composed of minimal pairs. Minimal pairs are arranged as follows: a word with meaning vs. a logatom (a word without meaning), e.g. *bałwan (a snowman) – pałwan*.

Memorisation of units. The tests examines the auditory memory. A given

number of verbal stimuli is presented and the subject has to point at the pictures corresponding to the stimuli in the same order as they were heard.

Memorisation and identification of linguistic structure. The aim of this test is to assess the ability of making use of the contextual and syntactic information in combination with segmental information in perceiving simple sentences. The test material consists of sentences of the same structure – the subject, the predicate and the object.

Recognition of phrases - Pan Ziemniak (Mr Potato). The aim of this test is to recognise individual words in a phrase which always begins the same way. In the three steps of the test the subject is asked first to draw the parts of Mr Potato's body, e.g. stomach or legs. In the next step the subject dresses Mr Potato. The subject is asked to put on Mr Potato e.g. gloves or shoes. Finally, the subject is asked to give Mr Potato different things, e.g. an umbrella or a balloon.

Recognition and intelligibility of complex phrases. The aim of this test is to assess the ability to recognise and comprehend speech by pointing at the correct key word. The tests are designed for children who are able to comprehend speech in open sets but who are unable to take a test requiring verbal responses. The tests are composed of lists of simple questions. In order to answer the question the subject points at the correct picture among three pictures presented on the computer screen.

Recognition and comprehension of continuous speech - a battery of thematic tests are designed for children who are able to comprehend speech in open sets but who are unable to take a test requiring verbal responses. In these tests the child does not have to hear precisely each part of the sentence, but he or she should be able to reconstruct the content of the speech.

The results of the first series of tests in this use case indicated that more flexibility would be provided by more extensive use of a speech synthesiser of higher quality than currently available. That result provided the motivation for the further research into ways of improving the speech synthesiser.

2.3.2 Use case: Test administration by perception testers (component #4)

The perception tests are designed for use by audiologists and speech therapists. They can be used by the audiologist in programming the cochlear implant, or by the speech

therapist as an achievement test. The set of speech perception tests is also useful teaching material and it can be used by parents to help their children work on their perceptive skills. The standard graphical user interface will need to be extended by manipulation options for synthesised voices. Figure 2 shows the scenario of the tests. During the testing procedure three subjects are involved: the child, the tester and the computer. A parent's presence during the tests is optional.

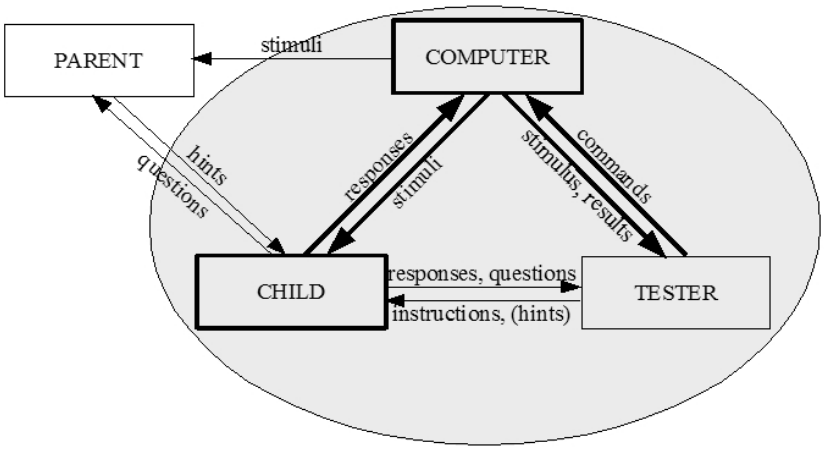

Figure 2: Test scenario showing communication relations between child, computer, tester and parent.

In the first stage of the testing procedure the tester provides the subject with instructions. If the subject understands the instructions, the tester runs the tests and the testing material appears on the computer screen. If the subject cannot understand the instructions, the test is terminated. The computer provides acoustic stimuli for the child, the tester and (if present) the parent. Then the child responds to the stimuli by pointing at a picture visualising the acoustic stimuli. If the child does not know what the stimulus is, he or she asks the tester or the parent questions. In principle, the tester is not allowed to give hints, but, for the purpose of the preliminary research (evaluating the tests), the testers may help the children with the tests if necessary. Similarly, the parents are asked questions by their children, and despite the fact that in principle they are also not allowed to give help, it is understandable that the parents help their little children with answers, and this is permitted for the present purpose of evaluating the tests. This kind of cooperation between the child, the tester and/or the parent is one of the main complicating factors in assessing the structure of the tests and the dialogue between the child and the computer.

All the responses given by the child to the computer are collected and the results

of the test are available on the computer screen to the tester. Finally, the tester notes down the results for future processing.

The current scenario requires the tester to be present during the testing procedure. At present, the tester's role is to choose a proper test for the subject and note down the results of the tests. But when the synthesised speech component is introduced, the tester will be able to create speech stimuli appropriate for the testee on the spot. The work with the test will be more interactive and the results on the subject's speech perception more detailed.

2.3.3 Use case: Test evaluation (components #5, #6)

The set of speech perception tests was evaluated by students of Linguistics at Adam Mickiewicz University. The evaluation of the tests started in September 2005 and went as follows:

1. In September and October 2005 the verification of the preliminary version of the set of the verbal tests and the set of tests with nonsense stimuli was carried out on Polish children with normal hearing. The two sets of tests were administered to 19 five-year-olds and 18 six-year-olds. The children's hearing was examined by audiologists. All the children had normal hearing and were normally developed.
2. In May and June 2006 the verification of the corrected and completed version of the set of verbal tests was conducted on Polish children with normal hearing. 14 four-year olds, 21 five-year-olds and 22 six-year-olds took part in the verification. The children's hearing was examined beforehand by audiologists. All the children, except one four-year-old, had normal hearing and were normally developed.
3. In June and July 2006 the set of verbal tests was verified on children with hearing aids and children with cochlear implants:
    1. Two Polish children with hearing aids sat some of the tests. One of the children was seven years old, the other was twelve years old.
    2. A group of 15 Polish children with a cochlear implant took some of the tests. The children were at different ages. The youngest children were 2,5

years old, the oldest were 11 years old. All the children were prelingually hearing-impaired. Only one girl lost her hearing at the age of five after having acquired a good command of speech.

Results in these scenarios can be compared in order to determine which manipulations of prosodic parameters lead to the best test results. The effectiveness of the set of speech perception tests was evaluated qualitatively by fourth-year students of Linguistics. In parallel to this, the tests were evaluated by audiologists. Note that the testers were concerned with evaluating the perception tests, not the cochlear implants themselves. The focus of the research was on evaluation of the level of efficiency, ergonomics, motivation and suitability of the tests for the subject. The testers evaluated many parameters. The relevant parameters for CCS development are as follows:

1. The intelligibility of instruction, picture and sound combinations used in the tests.
2. Dialogue between the child and the computer.

The problems discovered were:

1. Tests with nonsense stimuli:
    1. Recordings: Synthesised stimuli in the set of tests with nonsense stimuli were of poor quality.
    2. Interaction: The children had problems understanding the instructions to the tests with nonsense stimuli. The instruction were provided by the tester, they were not synthesised. But the little subject could hardly understand what was their task, because the tasks were not trivial and the teaching module was not very clear.
2. Tests with verbal stimuli:
    1. Recordings:
        1. Some sounds were very difficult to recognise, because of the speaker's fast speech rate.
        2. The pitch of the female voice was too low.
        3. The accentuation was not prominent enough for the purpose of some

tests.

4. Some sounds were segmented incorrectly.
5. Some sounds were missing.

2. Interaction:
    1. The dialogue between the testee and the computer needs improvement. Children sometimes did not know whether they gave a correct answer or not. They also looked at the testers or the parents for a sign of confirmation before giving the answer.
    2. If children with a cochlear implant could not understand the stimuli, they tried to read the word from the lips of their testers or their parents.
3. Scenario: There is no test including stimuli presented in noise.

For discussion of these results, see Bachan (2006). The results provided a specific set of requirements for CCS development:

1. Introduction of speech synthesis would avoid problems connected with speaker's pronunciation, e.g. very fast speech rate and inaccurate speech production.
2. Synthesised stimuli are very precise, flexible and easy to control.
3. Tests in which synthetic speech are used provide more detailed results.

2.3.4 Use case: Software evaluation (component #7)

The task for the software evaluation use case is to coordinate functional evaluation results from other components in the form of recommendations to the software developers. In practice, evaluation results may filter directly to the software developer, but in the ideal case the software evaluator will relate the evaluations to the original project goals before proposing software revisions and further development.

Based on the original project goals, some future directions for software development emerged from the evaluations:

1. Introduction of higher quality speech synthesis in order to correct the existing

synthesised stimuli and make speech stimuli dynamic and flexible.

2. Addition of a calibrated test in noise, preferably using speech synthesis.
3. Design and implementation of a database for stimuli and results.

The ideal situation would be if the results were sent via the Internet and stored in a database on a server. The data could wait there for future processing by speech therapists and audiologists. Figure 3 presents a model of such a database and its use.

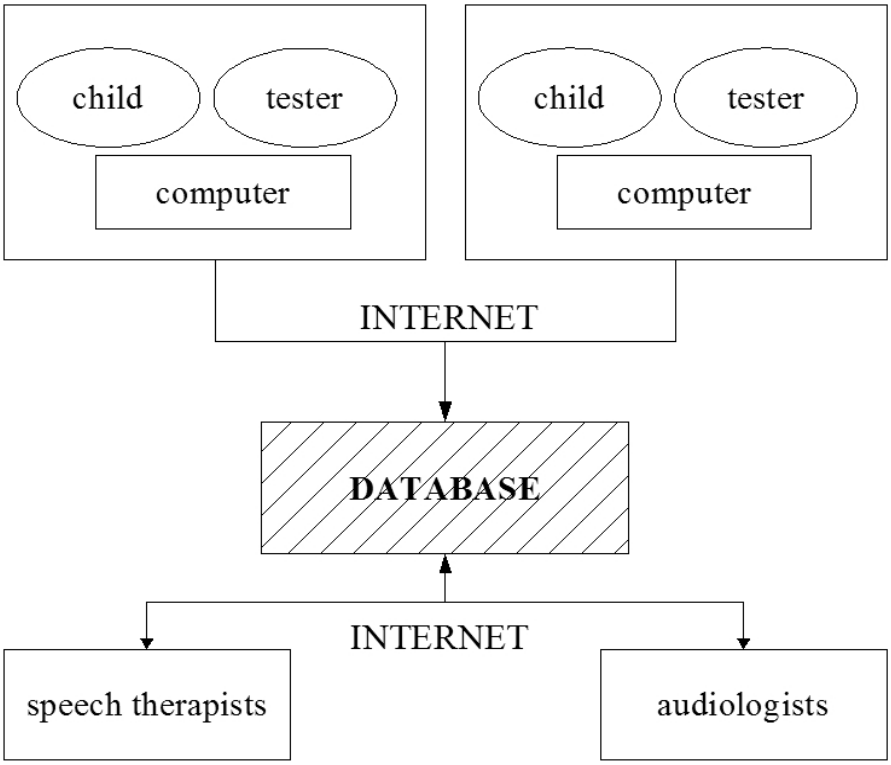

Figure 3: Networked database model with different access rights for different use cases.

In Figure 3 two testing procedures are illustrated in which a child, a tester and a computer take part. (In reality there could be more than one testing procedures.) The perception tests may be administered at the same place at a different time, or may take place at the same time in different places. The subject takes the perception tests in the tester's presence. The children's test results are sent automatically via Internet to a computer database, therefore the tester does not have to write down anything. The database stores all the children's test results. This kind of saving data prevents the situation in which the test results get lost or are damaged, and additionally assures that the data format is systematic. To the database speech therapists and audiologists have the access. The test results are stored in the database for future

processing by these two groups of specialists who can also put the results of their data processing into the database. It could boost cooperation between speech therapists and audiologists and provide an easy exchange of data.

2.4 Summary – requirements specification

The use cases defined above outline the motivation for implementation of a speech synthesis component to the speech perception tests. The requirements specification pointed out in this chapter are:

1. More flexibility of test stimuli is needed, therefore high quality speech synthesis component should be built in the set of speech perception tests.
2. Built-in speech synthesiser would make the tests interactive, allowing testers to create speech stimuli appropriate for individual testees.
3. Synthetic speech are totally controllable and precise, unlike human speech which depends on individual speaker's characteristics.
4. Tests with synthetic stimuli give more detailed judgements on speech perception.
5. The existing synthetic stimuli are of low quality, therefore a high quality speech synthesiser is needed to improve the sounds.
6. Calibrated tests in noise should be introduced; a speech synthesiser is a good tool to create such tests.

These requirements entail a development of a speech synthesis appropriate for this kind of testing. The choice is to use Close Copy Speech synthesis which gives synthetic speech of the best quality and allows easy modifications to the speech signal. The synthesised stimuli which were used in the battery of tests with nonsense stimuli did not sound like a human speech and therefore need to be considerably improved.

Synthesised speech stimuli are very precise, totally controlled, modifiable and therefore speech-like synthesised stimuli are the best choice for use in any kind of speech perception testing. Speech synthesis gives opportunity to create new stimuli without the need of recording an inflexible human speech. The use of synthetic speech renders the tests more precise and the stimuli become flexible.

CHAPTER 3 Overview of Text-To-Speech (TTS) systems and methods

3.1 What is a Text-To-Speech (TTS) system?

A text-to-speech (TTS) synthesiser is a computer system which converts written text into human-like speech. Ideally, a TTS system should be able to read any text which is input to the system, but in practice it is not easy to achieve and TTS systems still need improvements. Dutoit (1997) defines text-to-speech as "the production of speech by machines, by way of the automatic phonetization of the sentences to utter" (Dutoit 1997: 13). There are several different methods to synthesise speech. These methods are classified into two groups:

1. Parametric synthesis – synthesis by rule, e.g. formant synthesis, articulatory synthesis.
2. Concatenative synthesis, e.g. diphone synthesis, unit selection synthesis.

All these methods have their advantages and disadvantages, which will be discussed in this chapter. But it should be underlined that work on speech synthesis and the development of the methods listed above have helped to get better understanding of speech production, and psychological research making use of synthetic stimuli has made it possible to gain an insight into speech perception (Mattingly 1974: 2453).

3.2 Parametric synthesis

Parametric speech synthesis is a method in which a model of the articulatory or acoustic properties of the human vocal tract is constructed. First, a speech corpus is recorded. The corpus contains isolated words, frequently only CVC sequences,

which represent as many phoneme transitions and coarticulation effects as possible for the language studied. Then the speech corpus is analysed and the data are represented in parametric form. Next the rules are found and the parameters (e.g. the lip aperture or formant frequencies) of the model receive values depending on the speech sounds which the model is to produce (Dutoit 1997: 178-179). This method is also called synthesis by rule, because it "models phonetic and phonological rules in some nontrivial way" (Mattingly 1974: 2451) which are then used to synthesise speech.

Klatt (1987) says that "a synthesis-by-rule program constitutes a set of rules for generating what are often highly stylized and simplified approximations to natural speech. As such, the rules are an embodiment of a theory as to exactly which cues are important for each phonetic contrast" (Klatt 1987: 752).

3.2.1 Formant synthesis

Formant synthesis is a parametric approach which uses an acoustic model to produce speech. It does not make use of stored speech units. At runtime parameters such as formant and antiformant frequencies, fundamental frequency, bandwidths are varied over time to create a waveform of artificial speech. There may be as many as 60 parameters which must be manipulated to generate speech (Stevens & Bickley 1990: 63). However, only as few as three formants are required to synthesise intelligible speech and four or five formants are sufficient to generate high quality speech (Donovan 1996).

Formant synthesis is based on the source-filter model of speech production. "The source-filter theory states that the vocal tract can be modelled as a linear filter that varies over time. The filter (i.e. a set of resonators) is excited by a source, which can be either a simulation of vocal cord vibration for voicing, or a noise that simulates a constriction somewhere in the vocal tract. The sound wave is created in the vocal tract, then radiates trough the lips" (Styger & Keller 1994: 114).

One advantage of synthetic speech produced by formant synthesisers is that it is easily intelligible, even at very high speeds. Moreover, such a speech signal does not have the acoustic glitches which are often unavoidable in concatenative synthesis. Additionally, it is possible to switch voices from one speaker to another by adding

special rules in the rule database (Dutoit 1997: 180).

An example of a formant speech synthesiser is the KlatTalk system developed in MIT in 1983. In the KlatTalk system several different voices were provided to approximate the speaking characteristics of men, women, and children. Detailed data on formant frequencies were stored for only two voices, a man's and a woman's. Other male and female voices were created by scaling formant frequencies for different vocal tract sizes and by adjusting an extensive set of synthesis parameters concerned with the voicing source (Klatt 1987: 756).

Although a debate continues on whether a synthetic speech should sound human or not, one disadvantage which is concerned with the synthetic speech produced by a formant synthesiser is that the synthetic speech sounds robotic and can hardly be mistaken for human speech. Improving the naturalness of the artificial speech so that it sounds human is possible, however, the rules to do so have not been yet discovered (Dutoit 1997: 180).

3.2.2 Articulatory synthesis

In articulatory synthesis a model of the human speech production system is created. The movements of the human articulators and vocal cords are described in as many details as possible. For articulatory synthesis models of the jaw, lips, velum, tongue body, tongue tip, and hyoid bone are built (Klatt 1987: 756). These models of human articulators are moved towards target positions for each phoneme using rules. “The rules reflect dynamic constraints imposed upon the articulators by their masses and associated muscles” (Donovan 1996: 13). The models set the articulatory control parameters of the synthesiser which may be for example jaw aperture, lip aperture, lip protrusion, velic aperture, tongue tip height, tongue tip position, tongue height and tongue position.

The key to creating a good articulatory synthesiser is to gain deep knowledge of the dynamic constraints on the articulators. If the constraints are known, then it is possible to reach target articulatory shapes and positions of the dependent articulators. Because there are many ways to achieve the desired goal, a set of optimal rules must be selected for the system (Klatt 1987: 757).

There are two types of articulatory synthesisers: two dimensional (2D) and three

dimensional (3D). Although the human vocal tract is 3D, early speech synthesisers were developed based on the data derived from X-ray analysis of natural speech. This source of information does not provide sufficient data of the motions of the articulators during speech. Additionally, X-rays do not say anything about the masses or degrees of freedom of the articulators (Lammetty 1999: 28-29, who cited Klatt 1987). But modern techniques such as MRI (Magnetic Resonance Imaging) makes it possible to generate a 3D model of the moving vocal tract (Engwall 1998: 1).

3.3 Concatenative synthesis

Concatenative speech synthesis concatenates or "glues together" pieces of recorded speech, i.e. waveforms. The system is based on a database of recordings created for a single speaker. Then the database is segmented into short units which can be phones, diphones, triphones, half-syllables, syllables, words or other units (Jurafsky & Martin 2000: 274). Concatenative speech synthesisers have a very limited knowledge of the data on which they operate, their knowledge mainly concentrates on the speech segments which are to be chained (Dutoit 1997: 180).

The quality of synthetic speech produced by this method sounds very natural and is of great interest to speech engineers. Unfortunately, the fact that the synthetic speech is generated from smaller speech pieces can make audible glitches in the speech output. Therefore techniques such as *smoothing* are adopted to render the segment transitions as smooth as possible by minimising the discontinuity at the boundaries (Ng 1998: 10).

3.3.1 Diphone synthesis

Diphone synthesis uses diphones as speech segments. A diphone (or a dyad) is a unit that begins in the middle of the stable state of a phone and ends in the middle of the following one. The database recorded for one speaker contains all the possible diphones in the language. Using diphones for speech synthesis is very convenient since a single diphone contains the important phonetic transitions and coarticulations between phones. Usually, the number of diphones for a language varies from 1000-2000. This requires approximately 3 minutes of speech or 5 Mbytes of 16 bit samples at 16 kHz (Dutoit 1997: 187). The small size of the speech corpus makes diphone synthesis so popular.

The core of a diphone synthesiser is the diphone database (also called the *voice*). Creating the database must be well thought-out, because is has to include all the possible diphones in the language of concern. Creation of the diphone database is mainly achieved in four steps (Dutoit et al. 1996: 1395-1396):

1. Creating a text corpus:
    1. A list of phones, including allophones if possible, for a given language is prepared.
    2. Out of the list of phones a list of diphones is generated.
    3. A list of words containing all the diphones is created. Each diphone should appear at least once; diphones in such positions as inside stressed syllables or in strongly reduced (i.e. over coarticulated) contexts should be excluded.
    4. The key words are put in a carrier sentence.
2. Recording the corpus:
    1. The corpus is read by a professional speaker with monotonous intonation.
    2. The speech is digitally recorded and stored in digital format.
3. Segmenting the corpus:
    1. The diphones must be found in the corpus and annotated either manually or automatically by the means of automatic segmentator.
    2. The position of the border between the phones is marked.
4. Equalizing the corpus:
    1. The energy levels at the beginning and at the end of a segment is modified in order to eliminate amplitude mismatches – the energy of all the phones of a given phoneme is set to phones' average value (Dutoit 1997: 183).
    2. Pitch normalisation.

When the corpus is created it contains information in the following form: the name of the diphones, the corresponding waveforms, their durations and internal sub-splitting. Such created diphone database allows to modify the duration of one half-

phone without affecting the length of the other (Dutoit et al. 1996: 1396). Moreover, diphone synthesis enables to change the pitch of the segments.

One of the most popular diphone synthesisers is MBROLA (or MBR-PSOLA) which has been chosen for the purpose of the present study.

3.3.2 Unit selection synthesis

In unit selection synthesis, which is a special case of CCS synthesis, in a sense, naturally sounding speech output is produced by selecting sub-word units from a database of annotated natural speech (Black 2002). The speech database is large and there is a variable number of units from a particular class, for example phones. Each phone is marked with a duration and pitch values. The system has to find and select the best sequence of units based on questions concerning prosodic and phonetic context from all the possibilities in the database. The question which helps to find the target unit may be: Is the unit in a stressed syllable or is the unit at the end of a phrase? When the best units are selected, they are concatenated and speech output is produced (Black & Taylor 1997: 601).

In unit selection synthesis the approach taken is to use a large database of annotated speech, because it provides a large number of units with varied prosodic characteristics. This allows to synthesise speech which should sound more natural than the synthetic speech produced with a small set of controlled units, e.g. diphones (Hunt & Black 1996: 373, who cited Campbell & Black 1995). Although the database must be large to assure good synthesis, there are algorithms whose task is to reduce the size of the database. The method is called *pruning* and has two effects: firstly, units of bad quality are removed, secondly, units which are so common that there is no significant distinction between candidates are removed (Black & Taylor 1997: 602).

Additionally, because the trend of recoding larger and larger databases which would better cover the phonetic events led to creating very large databases whose storage may be troublesome, the new approach arose to select only the "right" data from the speech recordings. The two suggestions to do so are:

1. Model the acoustic space of a speaker – units which are acoustically distinct and frequent enough to count as worth storing are found out;

2. Utterances which best cover the required inventory are selected.

A database designed in such a way may consist of as few utterances as 500-1000 (Black 2002).

Another way to design a relatively small database is to build a specific database for a specific application. For specific application it is possible to predict what kind of expressions the system may be asked to synthesise or at least it is possible to define a subset of the language in question. Such information is the key to designing a relatively small database and cover well the application space (Black 2002). The techniques to built a domain specific speech synthesiser are described in (Black & Lanzo 2000).

When designing a unit selection synthesiser it is also important what kind of voice and speech style to choose for a specific domain. If a database is derived from a male voice, the synthetic speech sounds like a male. Additionally, if the database is designed to produce "calm" speech, it is not possible to make it shout (Black 2002).

One of the most popular unit selection speech synthesisers is Festival (Taylor et al. 1998). The Bonn Open Synthesis System (BOSS) is another unit selection speech synthesis system for whose purpose the corpus of recordings used in this study was created (Klabbers et al. 2001).

3.4 Summary – motivation for using diphone synthesis for CCS purposes

In this chapter various speech synthesis methods were presented, including formant and articulatory synthesis which make use of a model of acoustic or articulatory properties of the human vocal tract, and diphone and unit selection synthesis which use speech corpus of annotated recordings to produce speech. Having a large data of richly annotated recordings, each of the two latest methods would serve to develop Close Copy Speech synthesis. Although formant and articulatory speech synthesis methods give a very good insight into the way in which the human vocal tract works, they are not suitable for the present study, because the data required for these synthesis methods are not available, i.e. a model of a human vocal tract would have to be created.

The final choice was to be made between diphone synthesis and unit selection

synthesis. Both methods produce natural sounding speech, so the criterion to choose one of the methods was the flexibility of the speech units. Unit selection synthesis produces speech of high quality, with not many glitches at segment transition, however does not allow flexible prosody manipulations. The speech units are taken from the speech corpus without any extra processing. Theoretically, pitch and duration can be modified in unit selection systems, but the problem is to develop the rules which would do so. At present, a lot of work remains to be done in this respect. Taking this into account, diphone synthesis was chosen for developing the Close Copy Speech synthesiser. Diphone synthesis allows flexible prosody manipulation of pitch and duration at phoneme level, therefore was the most suitable for the present study.

CHAPTER 4 Requirements for speech synthesis

4.1 System requirements

The inputs required by the CCS synthesis system are as follows:

1. Source speech:
    1. source speech recordings,
    2. source database: annotated speech database.
2. Speech synthesiser:
    1. diphone database,
    2. synthesis engine.
3. Parametrisations of Close Copy Speech synthesis (PCCS synthesis) (not discussed in detail in this thesis).

The outputs to be produced by the CCS synthesis system are as follows:

1. Target pronunciation specification: specification table for input to speech synthesis engine.
2. Target acoustic output: produced by the speech synthesis engine.

An additional user interface for interacting with the system will be required for the PCCS synthesis system. There are three main sets of operations which users in one or more of the use cases will need to control in a user interface:

1. Duration warping: various linear or non-linear changes in the durations of phonemes in the utterance.
2. Frequency warping: various linear or non-linear changes in the frequency of

whole utterances or parts of utterances such as focussed syllables or nuclear tones.

3. Database management and stimulus presentation.

But these issues are not discussed in further detail here.

4.2 Available resources: recordings

A corpus of recordings[1] for a male voice was available from a speech synthesis development scenario. The texts were spoken by a professional speaker and the recordings were made in a professional recording studio. The sampling rate of the data in the available format is 16kHz in a standard WAV format. The texts for use in the synthesiser development consisted initially of a selection of 1400 sentences from the corpus of approximately 3240 utterances.

In the corpus of recordings five bases with different kinds of sentences are found:

1. Base A – 367 sentences with most frequent Polish consonant clusters:
    1. CCC, e.g. [blj]: Bibliotekarka zamknęła dzisiaj szkolną bibliotekę wcześniej. (The librarian closed the school library earlier today.)
    2. CCCC, e.g. [skSt]: Twoje litery są bezkształtne, dlatego musisz często ćwiczyć pisanie. (Your letters are shapeless, therefore you need to practise your handwriting often.)
    3. CCCCC, e.g. [fstfj]: Nie znam się na literaturoznawstwie. (I am not knowledgeable about literary studies.)
    4. CCCCCC, e.g. [mpstfj]: Nikt nie chce mówić o tym przestępstwie. (Nobody wants to talk about this crime.)
2. Base B – 114 meaningless sentences with all Polish diphones, e.g. W żądzy zejdę z gwoździa. (I will get down this nail in desire.)
3. Base C – 676 short sentences grouped into four, in each of which the same keyword appears. The keywords contain Polish triphones CVC in voiced

1 The author gratefully acknowledges the provision of this corpus by Grażyna Demenko (Principal Investigator of the Cochlear Implant Evaluation project).

context and in different intonation patterns:

1. at the beginning of a statement, e.g. Buzia jest całkiem ładna. (The face is quite pretty.)
2. in the middle of a statement, e.g. Taka sympatyczna buzia dużo znaczy. (This nice face means a lot.)
3. at the end of the statement, e.g. Najpierw zobaczyłem buzię. (First I saw the face.)
4. at the end of a question, e.g. Czy umyłeś buzię ? (Did you wash your face?)

4. Base D – (available) 200 sentences grouped into six, in each of which the same keyword appears. The keywords contain Polish triphones CVC in sonorant context and in different intonation patterns:
    1. at the beginning of a statement, e.g. Błogosławieństwa udziela ksiądz. (A priest gives the blessing.)
    2. in the middle of a statement, e.g.
        1. Pierwsze błogosławieństwo jest potrzebne. (First blessing is needed.)
        2. To nie jest błogosławieństwo tylko banały. (This is not a blessing, but banalities.)
    3. at the end of a statement, e.g. To nie są banały tylko błogosławieństwo. (These are not banalities, but a blessing.)
    4. at the end of a question, e.g. Czy dał błogosławieństwo? (Did he give the blessing?)
    5. at the end of an exclamation, e.g. Przecież tutaj jest napisane błogosławieństwo! (But here is written the blessing!)
    6. at the end of a continuation phrase, e.g. Wyraz błogosławieństwo – w języku polskim – niewiele znaczy. (The word blessing in the Polish language does not mean a lot.)
5. Base E – (available) 67 compound sentences with most frequent words from the vocabulary, e.g. Marek jest bardzo wrażliwym i czułym mężczyzną,

który wspaniale opiekuję się swoją rodziną. (Mark is a very sensitive and loving man who takes care of his family wonderfully.)

4.3 Available resources: annotations

Annotation of the recordings at phoneme level was performed automatically using the software tool CreatSeg (Demenko & al. 2006) and checked by trained phoneticians. Phonemic segments which were not correctly handled by the automatic segmentator were manually edited. Additionally, the annotations also contain prosodic information, based partly on functional judgements and partly on prosodic information. The annotation uses the following information types:

1. Sample serial numbers (column 1).
2. Phonemic/allophonic label tier (column 2):
    1. Labels for 40 phonemes. Table 1 shows a list of phoneme labels used in the annotation (Demenko et al. 2003: 85, cf. Jassem 2003: 103, 105).
    2. Stress and accent types (Demenko et al. 2006: 462):
        1. [%] rising accent realized by F0 rise on postaccented syllable/syllables or F0 interval between accented and postaccented vowels;
        2. ['] rising accent realized by F0 change (rise on accented syllable);
        3. ["] falling accent realized by F0 fall on postaccented syllable/syllables or F0 interval between accented and postaccented vowels;
        4. [&] falling accent realized by F0 change (fall on accented syllable);
        5. [|] rising-falling accents with rise-fall shape of F0 movement on accented vowel;
        6. [*] level accent realized by F0 interval between preaccented and accented vowels; near zero slope of fundamental frequency;
        7. [<] level accent realized only by differences in duration between preaccented, accented and postraccented vowels.
    3. Word and syllable boundaries (spaces indicate line breaks in the annotation files):

1. [#] prosodic word initial, e.g. [#j "e s t] (is);
2. [_#] orthographic word initial, e.g. [#z _#d a .l &e ./k a] (from far away);
3. [.] syllable word initial, e.g. [#d a. l &e ./k o] (far away);
4. [:] not clear pause, mark only on vowels, with very long part of very small intensity, e.g. [#t :o] (to) .

4. Four additional labels, including labels for paralinguistic information:
    1. [?] for a glottal stop,
    2. [#$p] for a pause,
    3. [#$j] for a segment such as a click or a sigh which is to be deleted for the purposes of speech synthesis. If [$j] is added to the first segment of a word, the whole word is to be deleted for the speech synthesis purposes, e.g. [#m] means the first segment of a word, [#$jm] means the first segment of a word which is to be deleted.
    4. [/] for a syllable segment which is to be deleted for the purposes of speech synthesis, for example because of creaky voice, not regular, very low F0, very small intensity, or background noise. E.g. [#d a. l &e ./k o] (far away) – the syllable [k o] will not be taken into account for the purpose of speech synthesis.

3. Prosodic tier (column 3) - Prosodic phrase boundary labels:
    1. [-5,.] Intonation on the first word in a sentence with falling accent F (or level accent L). In most cases it is used for declarative sentences or wh-questions. Mark on the first phoneme of the first word in the sentence.
    2. [-5,?] Intonation on the first word in sentence with rising accent R. It can be used in different complex sentences. Mark on first phoneme of the first word in the sentence.
    3. [5,.] Intonation on the last word in sentence with falling accent F (or level accent L). In most cases it is used for declarative sentences or wh-questions. Mark on first phoneme of the last word in the sentence.

4. [5,?] Intonation on the last word in sentence with rising accent R. In most cases it is used for yes-no questions. Mark on first phoneme of the last word in the sentence.
5. [5.!] Intonation on the last word in sentence with falling accent F. In most cases it is used for exclamatory sentences. Mark on first phoneme of the last word in the sentence.
6. [2,?] Intonation on the last word in the phrase with rising accent R. In most cases it is used for continuation phrase. Mark on first phoneme of the last word in the phrase.
7. [2,.] Intonation on the last word in the phrase with falling accent F (or level accent L). In most cases it is used in declarative phrases in complex sentences. Mark on first phoneme of the last word in the sentence.

Prosodic information is not needed for CCS synthesis, because it is extracted directly from the annotation files and the speech signal, therefore is deleted. Strictly speaking, only information about phonemes, phoneme's durations, i.e. sample serial numbers, and pauses are used for the MCCS and ACCS synthesis purposes. However, prosodic information can be taken into consideration later for Parametric Close Copy Speech (PCCS) synthesis or full speech synthesis system development where, together with prosodic information, grapheme-to-phonemes rules would be implemented (Stefen-Batogowa 1975). In principle, deleted paralinguistic information could be taken into consideration at a later stage, for example for synthesising elements of speech such as hesitation and sighs.

Table 1: SAMPA phoneme labels used in the corpus annotation.

BLF Polish modified SAMPA	*Orthography*	*Phonemic transcription*	*BLF Polish modified SAMPA*	*Orthography*	*Phonemic transcription*
p	pik	pik	i	kit	kit
b	byt	byt	y	typ	typ
t	test	test	e	test	test
d	dym	dym	a	pat	pat
k	kat	kat	o	pot	pot
g	gen	gen	u	puk	puk
c	kiedy	cjedy	@ - *schwa*		
J	giełda	Jjewda	m	mysz	myS
f	fan	fan	n	nasz	naS

BLF Polish modified SAMPA	*Orthography*	*Phonemic transcription*
v	wilk	vilk
s	syk	syk
z	zbir	zbir
S	szyk	Syk
Z	żyto	Zyto
s'	świt	s'fit
z'	źle	z'le
x	hymn	xymn
t^s	cyk	t^syk
d^z	dzwon	d^zvon
t^S	czyn	t^Syn
d^Z	dżem	d^Zem
t^s'	ćma	t^s'ma
d^z'	dźwig	d^z'vik

BLF Polish modified SAMPA	*Orthography*	*Phonemic transcription*
n'	koń	kon'
N	pęk	peNk
l	luk	luk
r	ryk	ryk
w	łyk	wyk
j	jak	jak
w~	ciąża	t^s'ow~Za
j~	więź	vjej~s'

The annotations are in the BOSS Label File (BLF) format, designed for the BOSS "Bonn Open Speech Synthesis" system (Klabbers et al. 2001). Table 2 shows the structure of the BLF annotation file. The file represents a three column matrix, with sample numbers in the first column, an allophonic representation including word and syllable boundary allophones and lexical stress types in the second column, and a prosodic boundary representation in the third column. The use of sample numbers and not time stamps makes additional knowledge of sampling rate metadata necessary. The table represents the first part of the Polish sentence *Na szczęście myśl o przeprowadzce była tylko chwilowa i Gosia będzie nadal z nami mieszkać*. (Fortunately, the idea of moving out was only temporary and Gosia will be still living with us.) from the corpus.

Table 2: Fragment of BLF file input resource.

Sample number (16 kHz rate)	*Segmental labels*	*Prosodic labels*
0	#$p	
5798	#n	-5,.
6863	a	
8008	#S	
9312	t^S	
10047	"e	
10880	j~	
11351	.s'	
12640	t^s'	

Sample number (16 kHz rate)	*Segmental labels*	*Prosodic labels*
13634	e	
14481	#$jm	
15613	|y	
16235	z'	
17214	l	
18843	#o	

In the phonemic/allophonic annotation label column, the following conventions are applied:

1. [#] encodes the beginning of a word, e.g. [#n] stands for a word-initial allophone of the phoneme /n/,
2. [.] encodes the beginning of a syllable, e.g. [.s'] stands for a syllable-initial allophone of the phoneme /s'/,
3. ["] denotes falling accent realised by F0 fall on postaccented syllable/syllables or F0 interval between accented and postaccented vowels, e.g. ["e] stands for the accented allophone of the phoneme /e/ with falling accent,
4. [#$p] stands for a pause ([#$p] is always inserted at the beginning and at the end of a sentence and can also appear in the middle of a sentence),
5. label [#$jm] is read as
    1. [#m] - word-initial allophone of the phoneme /m/,
    2. [$j] - a segment not to be used for the speech synthesis; the whole word is ignored for the purposes of speech synthesis.

In the prosody label column information about the type of utterance is represented:

1. [-5,.] indicates the beginning of a sentence with falling intonation.

For further information cf. section above and cf. Demenko et al. (2006).

4.4 Available resources: diphone database

The diphone database used in the study is the PL1 MBROLA Polish female diphone database[2] created under the free database access terms of the MBROLA project. The

2 Created by Krzysztof Szklanny and Krzysztof Marasek, whose work I gratefully acknowledge.

diphone database consists of 1443 diphones and contains 37 phonemes in standard Polish SAMPA notation. All the phonemes are listed in Table 3.[3] No Polish male diphone database is available for MBROLA.

Table 3: Polish SAMPA transcription used in the PL1 Polish female MBROLA voice.

PL1 Polish SAMPA	*Orthography*	*Phonemic transcription*	*PL1 Polish SAMPA*	*Orthography*	*Phonemic transcription*
p	pik	pik	i	kit	kit
b	bit	bit	I	typ	tIp
t	test	test	e	test	test
d	dym	dIm	a	pat	pat
k	kat	kat	o	pot	pot
g	gen	gen	u	puk	puk
f	fan	fan	e~	gęś	ge~s'
v	wilk	vilk	o~	wąs	vo~s
s	syk	sIk	m	mysz	mIS
z	zbir	zbir	n	nasz	naS
S	szyk	SIk	n'	koń	kon'
Z	żyto	ZIto	N	pęk	peNk
s'	świt	s'fit	l	luk	luk
z'	źle	z'le	r	ryk	rIk
x	hymn	xImn	w	łyk	wIk
ts	cyk	tsIk	j	jak	jak
dz	dzwon	dzvon			
tS	czyn	tSIn			
dZ	dżem	dZem			
ts'	ćma	ts'ma			
dz'	dźwig	dz'vik			

3 One of the differences between the PL1 MBROLA Polish female database phoneme set and the SAMPA Polish phoneme set used in the corpus annotation is that the former does not have /c/ and /J/ phonemes. However, these phonemes are not frequent in Polish, so there is no great data loss trying to replace them with phonemes available in the Polish diphone database (Łobacz 2002). Table 6, later in the article, shows all the differences between both sets.

CHAPTER 5 Design: Close Copy Speech (CCS) synthesis architecture

5.1 Text-To-Speech (TTS) synthesis – diphone synthesis

The standard components of regular text-to-speech synthesis are:

1. Natural Language Processing (NLP) module, which preprocesses and normalises an input text, produces phonetic transcription (phonetisation), together with a specification of prosodic features (pitch pattern, intensity and timing).
2. Digital Signal Processing (DSP) module, which transforms this data into speech, which may use uniform units such as diphones or corpus based weighted non-uniform unit selection.
3. Database of speech units such as diphones for the language in question.

The selected component MBROLA is a standard diphone synthesis engine: "MBROLA is a speech synthesiser based on the concatenation of diphones. It takes a list of phonemes as input, together with prosodic information (duration of phonemes and a piecewise linear description of pitch), and produces speech samples of 16 bits resolution (linear), at the sampling frequency of the diphone database used (it is therefore NOT a Text-To-Speech (TTS) synthesizer, since it does not accept raw text as input)."[4]

The MBROLA DSP component requires an input matrix containing phonemes, as well as specifications of duration and pitch modulation for each phoneme, but

4 MBROLA website, consulted 2006-11-30. I am grateful to the MBROLA team for this freeware application.

does not handle intensity modulation of the output. Figure 4 shows the architecture of a standard TTS diphone synthesis system with an MBROLA type synthesis engine. In Close Copy Speech synthesis, the NLP component is replaced with information from the speech corpus.

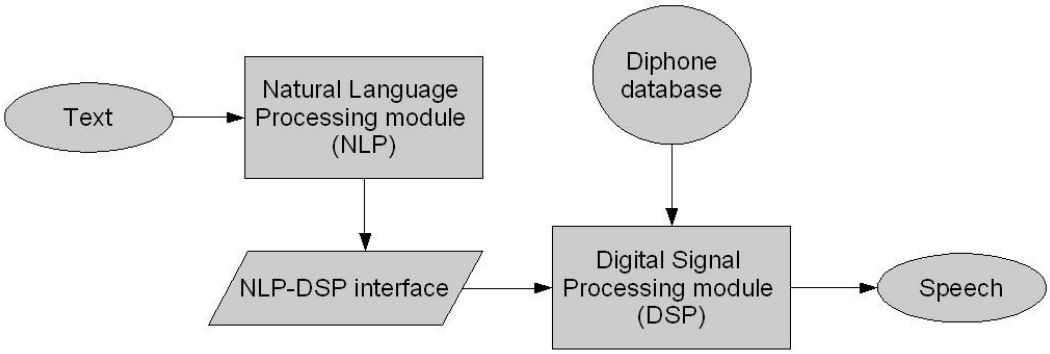

Figure 4: Text-To-Speech Synthesis architecture.

The architectures of manual and automatic Close Copy Speech synthesis procedures are identical but for the conversion component. In the MCCS synthesis procedure, the information from the original speech signal is transferred to a spreadsheet, and the mapping operations from the recordings (pitch extraction) and the annotations are performed manually. In the ACCS procedure, each of the manual operations is emulated by a software sub-component. The similar MCCS and ACCS synthesis architectures are shown in Figure 5.

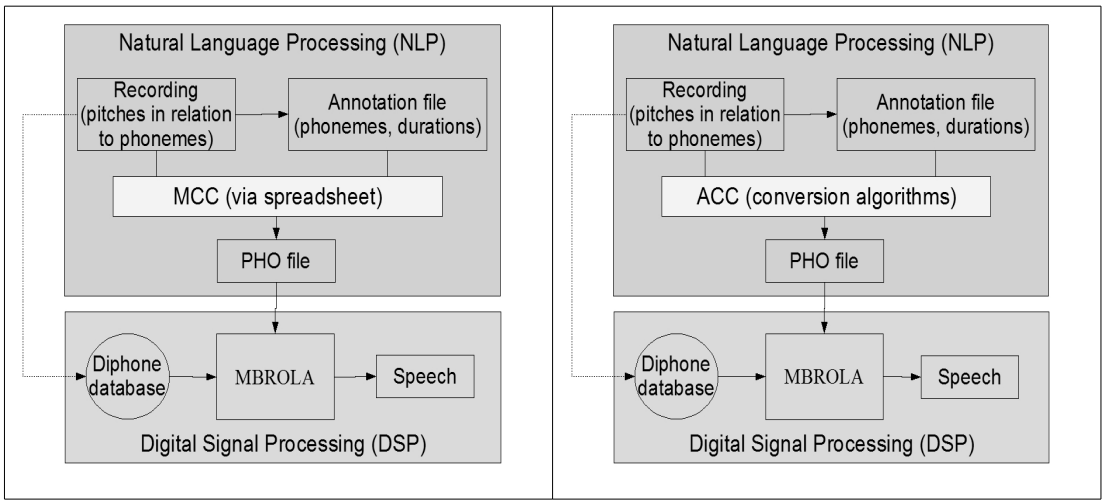

Figure 5: Schemata for similar architectures for Manual and Automatic Close Copy Speech synthesis.

In the following sections, the design and implementation of the MCCS and ACCS systems are described.

5.1.1 What is MBROLA type CCS diphone synthesis?

Close Copy Speech (CCS) synthesis is produced by the speech synthesis engine which has to "repeat utterances produced by a human speaker with a synthetic voice, while keeping the original prosody" (Dutoit, 1997: 134). For CCS synthesis, the standard MBROLA diphone synthesis architecture (Figure 4) is modified. The NLP component is replaced by an annotation file in which a transcription and a time stamp are aligned with the speech signal recording. The annotation and the recording together in principle include all the information which is needed for generating the specification table interface to the synthesis engine, which is normally produced by the NLP component. Consequently, in Close Copy Speech synthesis no input text is used. CCS synthesis makes use of recordings of real utterances and annotations derived from these recordings. In the annotation files, phonemes and their durations are stored. In the recordings, information about pitch in relation to the phonemes in the annotation files is found.

The module required for the kind of resynthesis selected for the development project is based on the MBROLA diphone synthesis model, which has the following structure:

1. Natural Language Processing (in TTS; in CCS generation from annotated recordings):
    1. Phonetisation: grapheme-to-phoneme conversion.
    2. Prosody generation: text parser for duration lookup and pitch assignment.
2. Specification table (PHO file) as NLP-DSP interface.
3. Speech synthesis component:
    1. Diphone database.
    2. MBROLA engine.
4. Audio (WAV file) output.

5.1.2 Phonotactics and diphone database set

One of the issues connected with speech synthesis with MBROLA is what happens if in the PHO file there is a sequence of phonemes which is not allowed to be present in

phonotactics for a given language. There may occur two errors of different kinds:

1. An unknown phoneme which is not present in a diphone database is being used.
    1. It may be caused when somebody does not know the phoneme set annotation convention of a diphone database in question and uses invalid symbols for transcribing an utterance.
    2. Somebody may deliberately use a phoneme which is not present in the language in question.

    In both cases an MBROLA error message appears: "Fatal error: Unknown recovery for x_1-x_2 segment." [x_1] stands for a phoneme which is recognised by a proper diphone database, [x_2] stands for the unknown phoneme.
2. An unknown sequence of phonemes is inserted in a PHO file. One can ask a question: Why is the sequence of phonemes not known? The answers can be:
    1. The sequence of diphones was not recorded. The creators of the diphone database omitted that sequence of phonemes, e.g. because the sequence is rare in the language.
    2. The sequence of phonemes is not present in the phonotactics of the language in question, therefore this sequence of phonemes is not included in the diphone database for the given language. If the reason for the absence of the sequence of phonemes is of this kind, then it implies that the transcription of an utterance in a given language used in the NLP-DSP interface is wrong. If it is the case, does MBROLA inform the user of an error made due to phonotactic rules? In the present version of MBROLA, the user is not informed of such an error.

The information about making a mistake in transcription due to using a sequence of phonemes which does not occur in a language in question may be a useful piece of information while learning transcription of a given language.

5.1.3 What is the NLP-DSP interface?

The central component for present purposes is the NLP-DSP interface which contains the pronunciation specification table produced by a TTS or CCS

component, and used as input by the synthesis engine to synthesise speech. In MBROLA the NLP-DSP interface is implemented by so-called PHO files, which will be discussed in a later section. The format specifies a table with three columns:

1. phonemes that are present in the sound to be produced,
2. duration of these phonemes,
3. pitch values represented by one or more pairs of numbers - the first number stands for the place of the pitch value in the phoneme, the second number is the pitch value itself.

The syntax of the specification table ST is defined as a sequence of one or more vectors SV, each with three components: the phoneme PH, the phoneme duration PD and the sequence of zero (for voiceless stretches) or more pitch pairs PP (in the prototype maximally one), consisting of pitch location PL and the pitch value PV:

```
<ST>  ::= <SV>+
<SV>  ::= <PH> <PD> <PP>*
<PP>  ::= <PL> <PV>
<PH>  ::= sampa_phoneme1 | ... | sampa_phonemen
<PD>  ::= millisecond_integer
<PL>  ::= pitch_location_percent
<PV>  ::= pitch_value_hertz
```

An illustration of the first five rows of the pronunciation specification table interface between the NLP and the DSP components is shown in Table 4; this example was derived from the corpus.

Table 4: Fragment of Specification Table (ST) for MBROLA PHO file.

PH phoneme (PL1 SAMPA)	*PD phoneme duration (msec)*	*PP pitch pair*	
		PL pitch location (%)	*PV pitch value (hertz)*
n	66	50	200
a	72	50	210
S	82	50	240
tS	45	50	310
e~	29	50	306

5.2 Implementation of diphone synthesis with MBROLA

5.2.1 What are MBROLA, Mbroli, phoplayer, diphone database?

The MBROLA tool package does not include a diphone database (voice). The package contains the following items:

Mbrola.dll:	Library file: MBROLA synthesis engine.
MbrPlay.dll:	Library file: user interface to the engine.
Mbroli:	A PHO file player with graphical user interface.
PhoPlayer:	A command line interface PHO file player.
Control Panel:	A control panel for managing the Mbrola Databases installed on the computer.
MbrEdit:	Another PHO player (written in VB, sources included).
C & VB source codes:	Interface to the DLLs.
Documentation	

Figure 6 shows a PHO file for a synthesised German sentence “Ich heiβe Jola” (“My name is Jola”). To create the file MBROLA software downloaded from the MBROLA Project website[5] was used. The procedure of creating the file is explained in the next section. In Figure 6 the following information is shown:

1. The first column shows phonemes present in the speech signal.
2. The next column shows the duration of the phonemes.
3. Next to the duration values there are pairs of pitch position and frequency values. The symbol “_” stands for a pause.

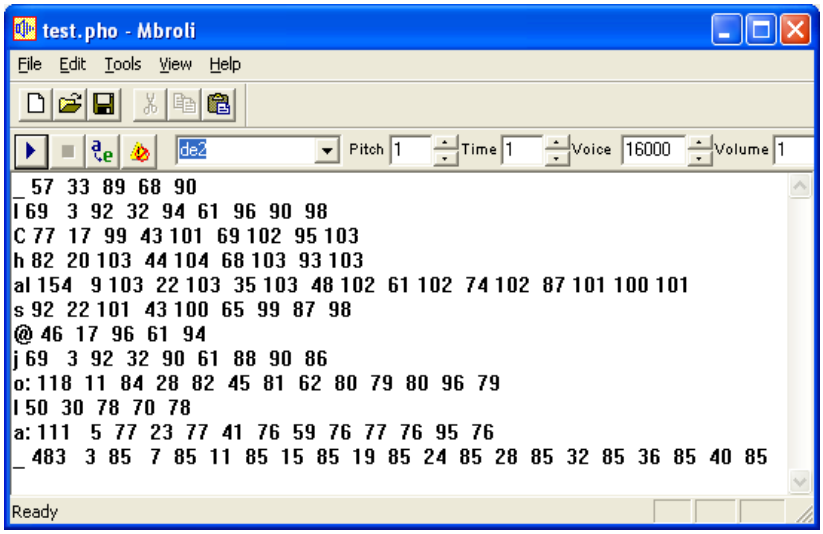

Figure 6: The PHO file for a German sentence "Ich heiße Jola".

5 MBROLA website, consulted 2006-10-15. I am grateful to the MBROLA team for this freeware application.

E.g. The first phoneme “I” lasts 69 ms. At the 3% of its duration F0 reaches the value of 92Hz. At position 32% the value of F0 is 94Hz. The value keeps rising and at 61% the pitch is 96Hz and finally F0 reaches 98Hz in the position of 90%. “de2” in the box at the tool bar says that a diphone database “de2” is being used (“de2” stands for a German diphone database for a male voice from the MBROLA website).

5.2.2 Implementation of the NLP-DSP interface as MBROLA PHO file

5.2.2.1 Automatic PHO file production with the use of an NLP module

If an NLP module is available for a given language, production of PHO files can be done automatically. To create a PHO file the following steps have to be taken:

1. Download and install MBROLA. (MBROLA is found on the the MBROLA Project website.)
2. Download and install a diphone database for the language in which a speech synthesis is going to be conducted. (All available diphone databases are stored on the official MBROLA Project website.)
3. Find, download and install an NLP module for the language in which the speech synthesis is going to be conducted, if available. This means that the diphone database and the NLP module must be designed for the same language. For the German language the NLP engine can be downloaded from IKP Forschung: Phonetik *Txt2Pho* website[6], for example. If an NLP module is not available then the module must be developed.
4. Write into a TXT file a text to be transformed into a speech signal.
5. Convert this TXT file into a PHO file with the use of the NLP module.
6. Play the resulting PHO file with the Mbroli or phoplayer application. (The new PHO file is created in the same folder in which the TXT file was saved.)

6 The IKP Forschung website, consulted 2006-10-15. I am grateful to the author of *txt2pho* software, Thomas Portele, for this application.

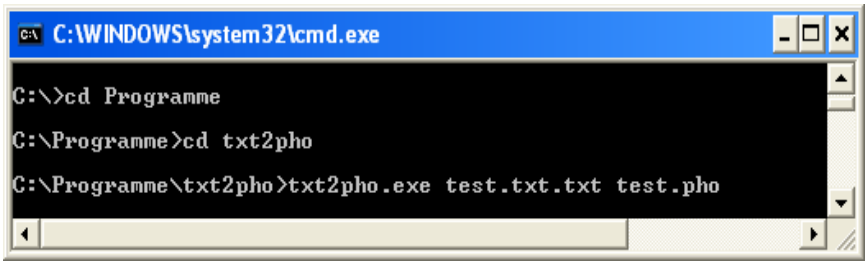

Figure 7: Conversion of a TXT file info a PHO file.

Figure 7 shows the procedure of conversion a TXT file named test.txt.txt into a PHO file named test.pho which is presented in Figure 6. A Natural Language Processing module for German called *txt2pho* is used. The file test.txt.txt is saved in the txt2pho folder, therefore the new file is created in the folder txt2pho. The command *txt2pho.exe test.txt.txt test.pho* means "Convert the file test.txt.txt into test.pho with the use of *txt2pho.exe* (the NLP module for German)."

5.2.2.2 Manual PHO file production without the use of the NLP module

Unfortunately, NLP modules are not available for all the diphone databases available. If the NLP module is not available, which is the case for Polish, it is still possible to make MBROLA speak. To do so a PHO file must be created manually. Creating a PHO file is as simple as that (with no guarantee for prosodic authenticity):

1. Transcribe a sentence to be synthesised. Make sure that the transcription uses only the symbols used by the diphone database which is to be used. (The transcription symbols can be found in the properties of the diphone database – Go to MBROLA Tools in the programme menu. Open the Control Panel. Choose the diphone database which is going to be used and check the properties.)
2. Open an empty PHO file (Go to MBROLA Tools and open Mbroli.)
3. Write the transcription as a column of phonemes.
4. Add the value of the duration to each of the phonemes.
5. Add a pair or pairs of pitch position and frequency values for the voiced phonemes. The voiceless phonemes do not require pitch pairs.
6. Run Mbroli, i.e. play the new-made PHO file with the Mbroli application.

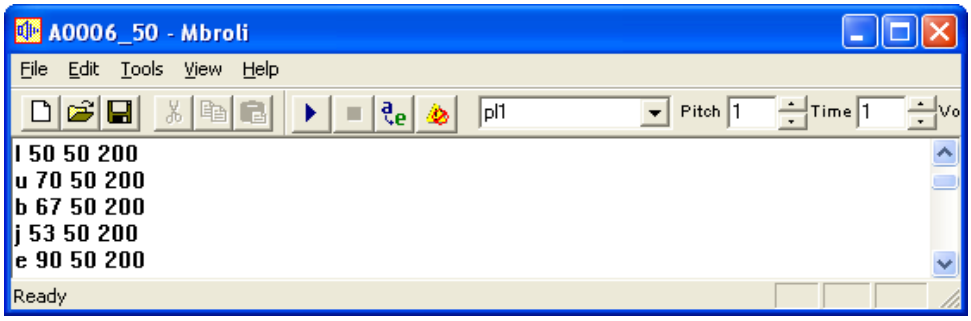

Figure 8: A PHO file for a monotonous voice.

In the simplest form the PHO file may look like the one presented in Figure 8. The PHO file shows the production of a Polish word “lubię” (“I like”). The phonemes match the transcription symbols used in the pl1 database (“pl1” stands for a Polish diphone database with a female voice, see the box on the tool bar). The duration values are made up, but correspond to the durations that may be produced in a real speech by a human being.. The pitch value is the same for all the phonemes and at the position of 50% adopts the value of 200Hz.

This application of the MBROLA system is valuable for didactic purposes, especially for systematic manual manipulation of the interface file (PHO file) parameters.

CHAPTER 6 Manual Close Copy Speech (MCCS) synthesis

6.1 What is MCCS synthesis?

Manual Close Copy Speech (MCCS) synthesis with an MBROLA type diphone synthesis is a process of manually creating pronunciation specification tables (NLP-DSP interfaces, implemented as PHO files), making use of recorded and annotated real utterances, and synthesising the pronunciation specification tables using an appropriate voice (diphone database). The voice may be created from the annotated utterances, in the ideal case, or may be an independently created voice, as in the case of the present study. The human copier therefore emulates the Natural Language Processing front end to a speech synthesis engine. The speech and annotation information is input by manual operations into the Manual Close Copy Speech (MCCS) synthesis procedure. The output of the MCCS synthesis procedure is a pronunciation specification table which, together with a diphone database, constitutes the input to the synthesis engine, which converts the specification table into speech using the diphone database. The acoustic output is a speech file.

Figure 9 visualises the Manual Close Copy Speech synthesis process. The Figure shows that to synthesise speech two modules are needed: the Natural Language Processing module and the Digital Signal Processing module. In the Manual Close Copy Speech synthesis no input text is used. The synthesis makes use of recordings of real utterances and annotations which are derived from these recordings. In the annotation files phonemes and their time-stamps are stored. In the recordings information about pitches in relation to the phonemes present in the annotation files is found. The information about phonemes, time-stamps and pitch is

input to the Manual Close Copy (MCC) procedure.

The output of the MCC, together with the diphone database, is input to the synthesis engine, which converts the PHO file into speech using the diphone database which may be created from the annotated recordings or may be taken from an external source. The output of MBROLA is a speech file in WAV format. The whole process is carried out in the Digital Signal Processing module.

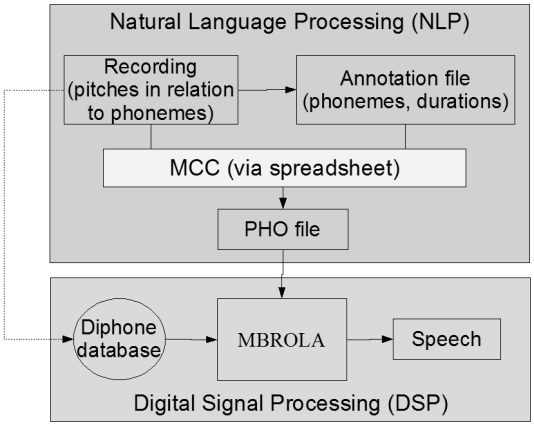

Figure 9:Scheme for MCCS synthesis.

6.2 Mismatches and format preprocessing

The specification table required by the speech synthesis engine when used with the available Polish diphone database resource differs from the table provided by the BLF Polish resource. This incompatibility has several aspects, for which format conversion tools need to be specified. The incompatibilities are listed in Table 5.

Table 5: Polish annotation, diphone database and pronunciation specification table (NLP-DSP interface) conventions

Polish annotation	*Diphone database*	*NLP-DSP interface*
sample numbers	-	durations (msec)
positional allophones	phonemes	phonemes
BLF phoneme set	PL1 phoneme set	PL1 phoneme set
syllable boundaries	-	-
word boundary types	-	-
pauses	pauses	pauses
prosodic annotation	-	-

The boundaries and the stress markings are not usable in the current configuration and are deleted, but will be considered at a later stage for prosody

parametrisation. The SAMPA phoneme set and notation was given by the available diphone database in the pre-processed input format, and differs from the phoneme set used in the corpus annotation. The correspondences are shown in Table 6.

Table 6: Mismatches between BLF and PL1 SAMPA.

BLF SAMPA annotation labels	*PL1 SAMPA symbols*
p	p
b	b
t	t
d	d
k	k
g	g
c	-
J	-
f	f
v	v
s	s
z	z
S	S
Z	Z
s'	s'
z'	z'
x	x
t^s	ts
d^z	dz
t^S	tS
d^Z	dZ
t^s'	ts'
d^z'	dz'

BLF SAMPA annotation labels	*PL1 SAMPA symbols*
i	i
y	I
e	e
a	a
o	o
u	u
@ - *English schwa*	-
-	e~
-	o~
m	m
n	n
n'	n'
N	N
l	l
r	r
w	w
j	j
w~	-
j~	-

A further mismatch occurs between the Polish annotation interface (BLF) and the NLP-DSP interface (PHO) formats for time specification. The BLF format includes sample numbers, while the PHO format requires durations. In order to calculate durations, sampling rate metadata information (16 kHz) is required. The formula for bridging the gap is $(samplenumber_i - samplenumber_{i-1}) / samplingrate$.

Perhaps the most crucial mismatch is between the corpus, which is recorded using a male voice, and the diphone database, which is derived from a female voice. This requires a pitch re-adjustment. Currently the trivial formula $pitch_{female} = 2 * pitch_{male}$ is used, but parametrisations with more complex formulae

incorporating a baseline are being developed. In the long term, an annotated corpus based on a female voice is required, as well as diphone databases based on male voices.

6.3 MCCS synthesis system implementation

The BLF and PHO format specifications do not match, as already indicated in the design specification. Nevertheless, the required information is implicit in the annotation file, and the annotation file may be mined for this information. For the purpose of the MCCS procedure, a spreadsheet was designed in order to convert BLF format into PHO format. In order to map the BLF format into the spreadsheet table, a pre-processing step is necessary: the three different columns (the sample numbers, annotation labels and phrase intonation labels) are placed in a CSV-formatted file which is opened as a spreadsheet table. The spreadsheet software used is OpenOffice Calc. The further steps required for conversion into the PHO format are detailed in Table 7 and described below.

The columns in Table 7 contain the following information:

1. Running BLF label indices (not in original BLF format).
2. BLF format: sample number.
3. BLF format: phonemic/allophonic annotation labels.
4. BLF format: prosodic labels.
5. Conversion of sample numbers to time-stamps (msec): division of sample numbers by sampling rate (16kHz), i.e. *sample-number*/16.
6. Phoneme durations: $duration(cell_i)$ - $duration(cell_{i-1})$, i.e. the value of the preceding cell is subtracted from the value in each cell.
7. PHO format: Polish Voice SAMPA phoneme notation, with BLF characters replaced by Polish Voice SAMPA characters. Phonemes whose characters are different from the ones used by the diphone database are converted into the Polish Voice SAMPA characters. All the characters which do not match characters used by the diphone database will not be recognised and the synthesis will not work.

8. PHO format: rounded (integer) phoneme duration values.
9. PHO format: Pitch Location: at 33% of phoneme length.
10. PHO format: Pitch Value in Hertz (*ad hoc* adaptation to female Polish Voice: multiplication by 2).
11. PHO format: Pitch Location: at 66% of phoneme length.
12. PHO format: Pitch Value in Hertz (*ad hoc* adaptation to female Polish Voice: multiplication by 2).
13. Original male voice pitch values at 33% (extracted manually using WaveSurfer software).
14. Original male voice pitch values at 66% (extracted manually using WaveSurfer software).

Table 7: Spreadsheet for BLF to PHO format conversion.

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*
	Samples	*BOSS 2006*	*Prosody*	*msec = Samples / 16*	*Duration (msec)*	*Polish Voice SAMPA*	*Duration – rounded*	*Pitch Location: 33%*	*Pitch Value: 33%*	*Pitch Location: 66%*	*Pitch Value: 66%*	*Original Pitch Value – male: 33%*	*Original Pitch Value – male: 66%*
1	0	#$p		0	70	_	70						
2	1120	#v	-5,.	70	95,44	v	95	33	174	66	226	87	113
3	2647	y		165,44	64,56	I	65	33	218	66	194	109	97
4	3680	.g		230	50	g	50	33	188	66	228	94	114
5	4480	l		280	36,63	l	37	33	234	66	252	117	126
6	5066	`o		316,63	83,38	o	83	33	262	66	284	131	142
7	6400	n		400	46,81	N	47	33	286	66	268	143	134
8	7149	.d		446,81	31,88	d	32	33	250	66	270	125	135
9	7659	a		478,69	51,31	a	51	33	256	66	236	128	118
10	8480	Z		530	122,94	Z	123	33	196	66	186	98	93
11	10447	#d^z'		652,94	67,06	dz'	67	33	182	66	220	91	110
12	11520	i		720	30	I	30	33	226	66	228	113	114
13	12000	.s'		750	120	s'	120	33	230	66	260	115	130
14	13920	a		870	60	a	60	33	238	66	216	119	108
15	14880	j		930	30	j	30	33	210	66	212	105	106
16	15360	#j		960	30	j	30	33	212	66	210	106	105

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*
	Samples	*BOSS 2006*	*Prosody*	*msec = Samples / 16*	*Duration (msec)*	*Polish Voice SAMPA*	*Duration – rounded*	*Pitch Location: 33%*	*Pitch Value: 33%*	*Pitch Location: 66%*	*Pitch Value: 66%*	*Original Pitch Value – male: 33%*	*Original Pitch Value – male: 66%*
17	15840	a		990	80	a	80	33	208	66	186	104	93
18	17120	g		1070	62,94	g	63	33	210	66	204	105	102
19	18127	#Z	5,.	1132,94	93,56	Z	94	33	200	66	200	100	100
20	19624	&a		1226,5	123,5	a	124	33	180	66	138	90	69
21	21600	./b		1350	62,94	b	63	33	136	66	184	68	92
22	22607	a		1412,94	167,06	a	167	33	148	66	146	74	73
23	25280	#$p		1580									

Figure 10 shows the process of manually extracting the pitch value at the position of 66% for the phoneme /o/ in the recording of a male voice. The value of the pitch is shown at the bottom on the left. In this case the pitch is 142Hz.

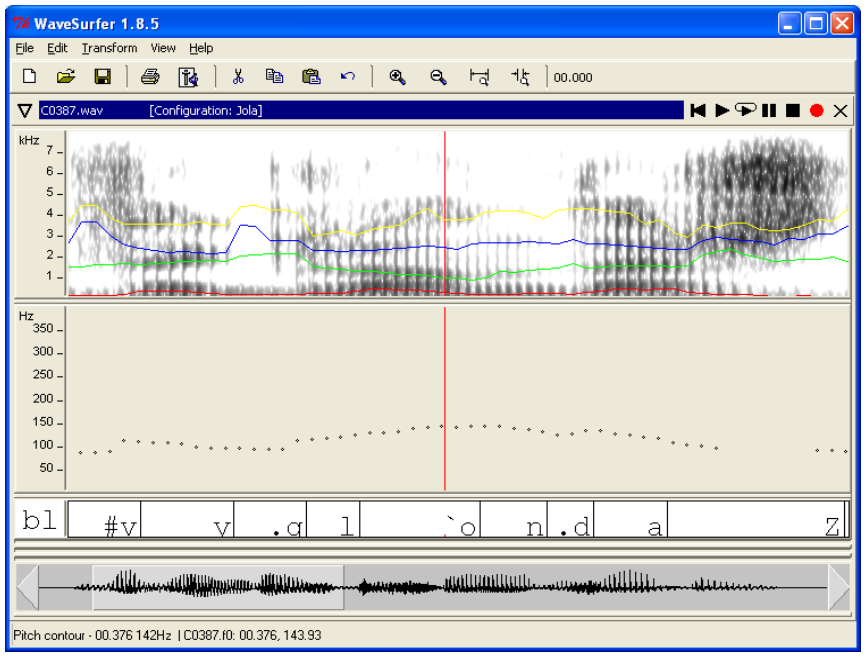

Figure 10: Extraction of the pitch in the position of 66% for the phoneme /o/ using WeveSurfer.

If the spreadsheet table is filled in, the creation of a new PHO file is easy to complete. The only steps to be taken are:

1. Open a new PHO file.
2. Mark the columns 7-12 (without the content of headings and the first and the

last row connected with the pauses), copy the columns.

3. Paste the columns 7-12 into the new PHO file.
4. Check if you are using a proper voice database.
5. Run Mbroli - play the new-made PHO file with the Mbroli application.
6. Save the PHO file.

The correct PHO file for the sentence C0387 is is shown in the Figure 11.

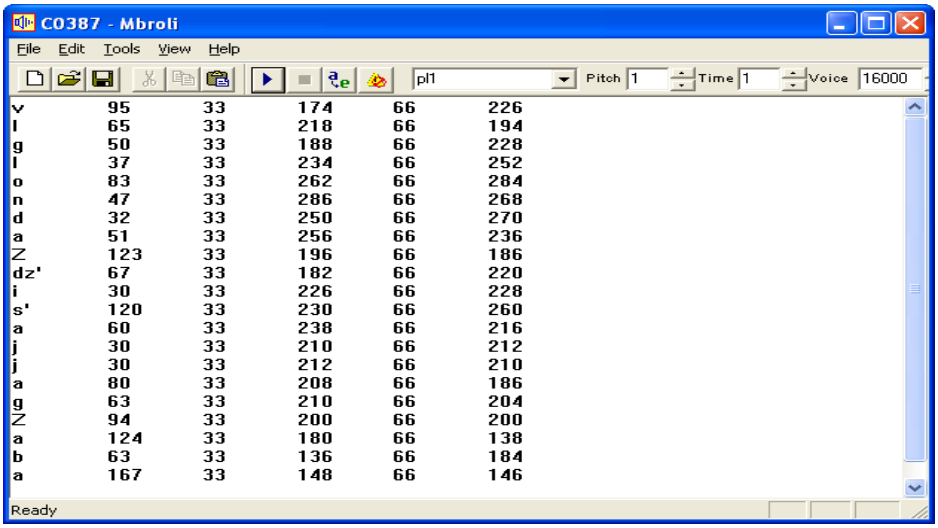

Figure 11: Manually close copied PHO file for the sentence C0387.

CHAPTER 7 Automatic Close Copy Speech (ACCS) synthesis

7.1 What is ACCS synthesis?

Automatic Close Copy Speech (ACCS) synthesis is similar to Manual Close Copy Speech (MCCS) synthesis, except that the transfer of parameters from the original signals and annotations is performed automatically by conversion algorithms. ACCS synthesis emulates MCCS synthesis and therefore should be as good as MCCS synthesis which gives synthetic speech of the best possible quality for a given synthesis engine.

Figure 12 shows the ACCS synthesis system design. The Perl script which carries out all the conversions is in the Appendix B.

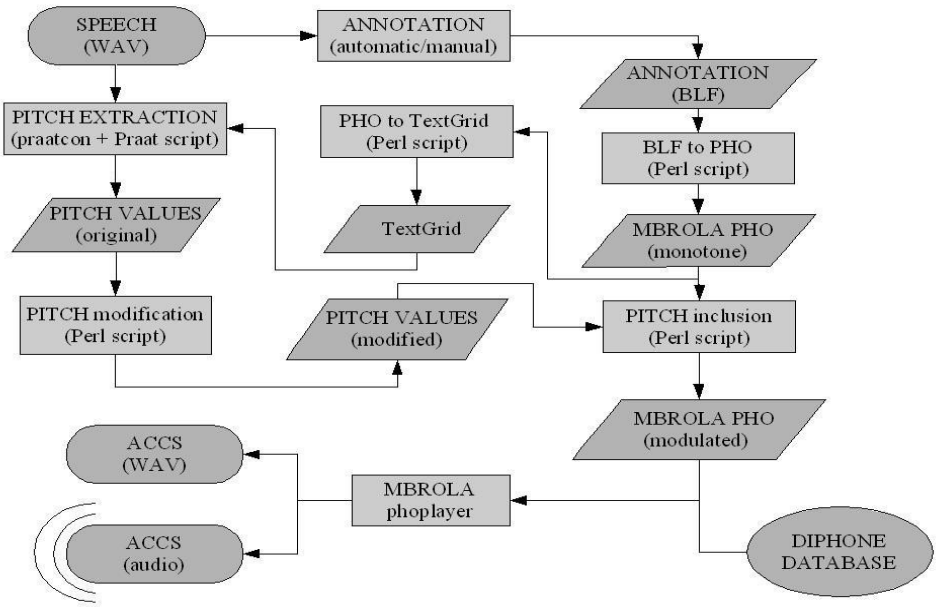

Figure 12: Modular structure of the ACCS system.

7.2 Components of the ACCS synthesis system

ACCS synthesis is a complex process of converting an annotation file into a sound file with preserving the pitch pattern which occurs in the speech file for which the annotation file is made. The components of the ACCS synthesis system are:

1. Speech information input:
    1. speech recordings,
    2. time-aligned annotations of speech recordings in BLF format.
2. Speech synthesiser:
    1. diphone database,
    2. synthesis engine.
3. Pitch extraction script:
    1. BLF to MBROLA PHO format conversion procedure,
    2. MBROLA to TextGrid (Praat format) procedure,
    3. pitch extraction (calling Praat script),
    4. inclusion of pitch values in MBROLA PHO file,
    5. synthesis of PHO file with MBROLA engine.

The ACCS synthesis system whose aim is to produce sound which "best mimics the natural speech presented at its input" (Dutoit, 1997: 193).

Figure 13 shows the architecture of the ACCS synthesis.

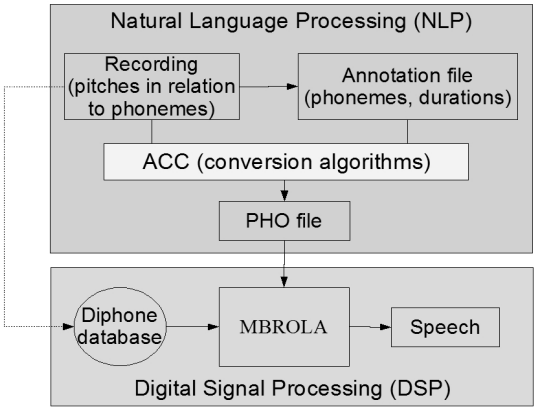

Figure 13: The architecture of ACCS synthesis system.

7.3 ACCS synthesis development procedure overview

The ACCS synthesis requires several conversion steps. The overall implementation architecture is shown in Figure 14. Since the Praat script for extracting the pitch velues requires a different format (TextGrid), the BLF sample number and phoneme notation were converted into both MBROLA PHO format and Praat TextGrid format. Three options were considered for this:

1. Conversion of BLF directly into TextGrid format.
2. Conversion of PHO format into TextGrid format, i.e. indirect conversion of BLF into TextGrid format, because BLF had already been converted into PHO format.
3. Conversion of BLF format into a generic XML format (TASX, cf. Gut & Milde 2003), for which a library of functions, including TASX to Praat, already exists.

Both option 2 and option 3[7] were implemented. The implementation of option 2 is straightforward. The implementation of option 3 is more re-usable, but depends on an additional Java environment and the Saxon XML engine, and is therefore more complicated than necessary for the prototype.

Figure 14 presents the overall ACCS synthesis implementation. The explanation of the figure is to be found in the following sections.

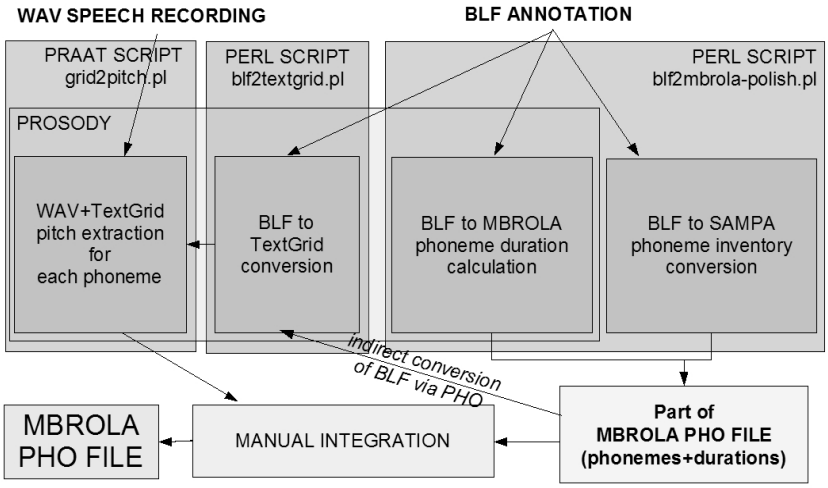

Figure 14: Schema of Automatic Close Conversion Speech synthesis.

7 I am grateful to Thorsten Trippel for his help and his software which made it possible to convert BLF to TextGrid automatically.

7.3.1 Automatic BLF phoneme set to Polish Female Voice (diphone database) phoneme set conversion

The main problem which appeared in the process of synthesising speech using the Close Copy Speech synthesis method was that phoneme set used for annotating recorded utterances (BLF SAMPA annotation phoneme set) was not the same as the phoneme set used by the diphone database for the Polish language (PL1 SAMPA phoneme set). Mapping most of the BLF annotation labels used by the Polish Female Voice (the diphone database) was not difficult. However, in some cases adapting the phoneme set used for annotation to the diphone database phoneme set was very tricky. The problem was caused by [ew~], [ow~], [ej~] and [oj~] sequences of phonemes in the BLF inventory, because those phonemes were equivalent to [e~] and [o~] in the diphone database inventory. The sequences of phonemes [e] or [o] followed by [w~] or [j~] must be replaced by one segment [e~] or [o~], depending on the context. Additionally, the duration of the [e~] or [o~] phoneme is a sum of the durations of [ew~], [ow~], [ej~] or [oj~]. Figure 15 shows a flow chart for conversion of the BLF SAMPA annotation phoneme set into the PL1 SAMPA phoneme set and creating PHO format from a BLF format.

Figure 15 presents the operations to be performed by the program while converting a BLF file into a PHO file. The input to the program is a BLF file. The program reads the first line of the file, makes the conversion, and prints the converted line in PHO format into the PHO file. Then the program reads another line, makes the conversion, prints the converted line in PHO format into the PHO file, etc. When all the lines are read, converted and printed into the PHO file, the PHO file is closed and it can be later input to MBROLA for synthesising speech. A detailed description of the algorithm visualised in Figure 15 follows.

1. The program opens a BLF file.
2. The program reads the first line if the file is not empty. The phoneme from the line is put into the $_ variable.
3. The program checks if $_ is empty.
4. If the $_ variable is empty, it means that either the BLF file is empty or the program has already read all the lines. Then BLF file is closed.

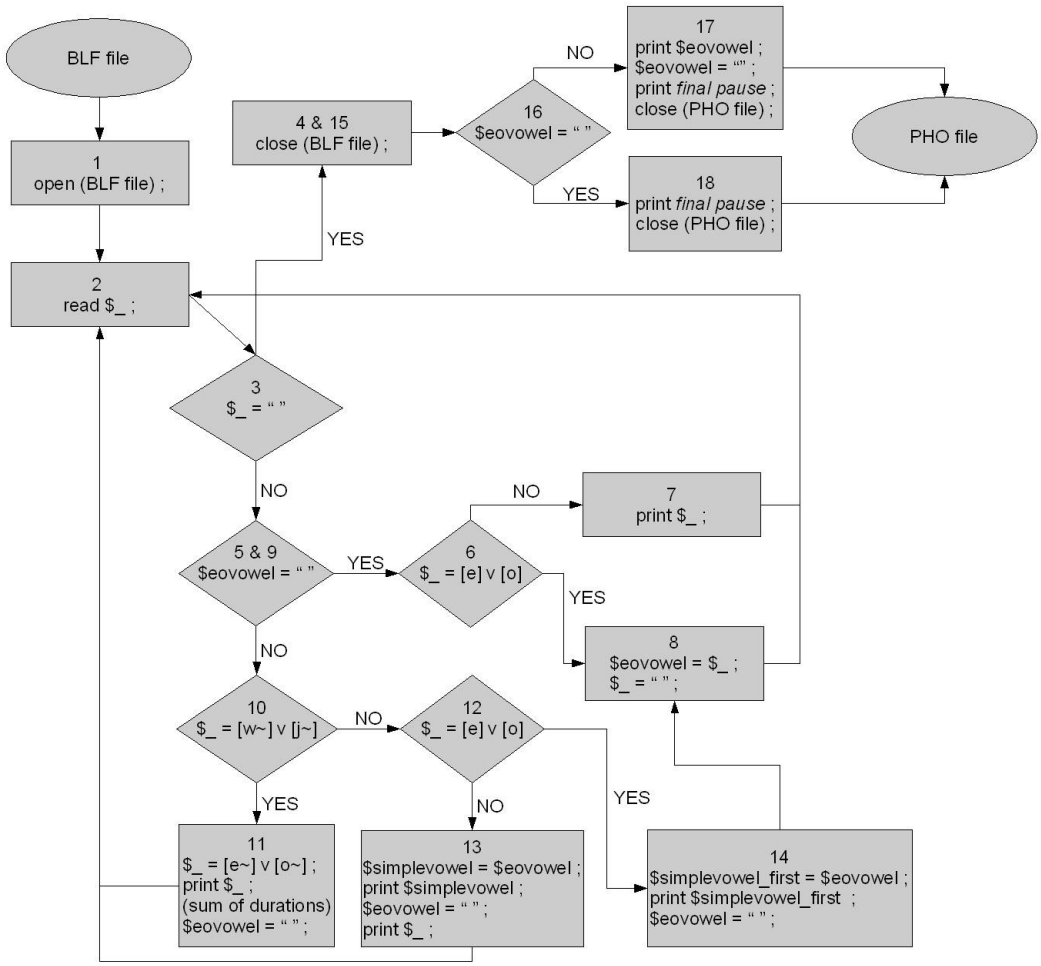

Figure 15: The schema for creating a PHO file from a BLF file with regard to conversion of BLF SAMPA annotation phoneme set into PL1 SAMPA phoneme set.

5. If the line is not empty, then the program puts the values from the line into variables, and checks if the variable $eovowel is empty.
6. If the $eovariable is empty, then the program asks if the variable $_ is equal to the vowel [e] or equal to the vowel [o].
7. If $_ does not contain [e] or [o], then it prints the value of the $_ variable to the PHO file and reads another line from the BLF file.
8. If $_ contains [e] or [o], then it does not print anything, stores the value of the $_ variable in the $eovowel variable and reads another line from the BLF file.
9. Then a new line is read if there is another line, i.e. if all the lines have not been read already. If there is another line, then the program checks if the $eovowel variable is empty.

10. If $eovowel is not empty, then it asks if the $_ variable contains [w~] or [j~].
11. If the $_ contains [w~] or [j~], then [w~] or [j~] is converted into [e~] or [o~], the duration of the new (current) phoneme is calculated by adding the duration of the previous phoneme, $eovowel, and the current phoneme, $_. Then the value of the $_ variable is printed, i.e. [e~] or [o~]. The $eovowel is emptied and the program reads a new line from the BLF file.
12. If $eovowel is not empty, and the $_ variable does not contain [w~] or [j~], then the program asks if the value of the $_ variable is equal to [e] or [o].
13. If $_ is not equal to [e] or [o], then the program treats the previous vowel [e] or [o] (because $eovowel is not empty) as a simple vowel and stores the vowel in the $simplevowel variable. Then the value of the $simplevowel variable is printed, as well as the phoneme stored in the $_ variable. The $eovowel variable is emptied and the program reads a new line from the BLF file.
14. If $eovowel is not empty, and the $_ variable is equal to [e] or [o], then the program knows that the first vowel stored in the $eovowel variable was a "simple vowel", puts the value of the $eovowel variable into a new variable called $simplevowel_first, prints the value of the $simplevowel_first, empties the $eovowel variable in order to put the value of the $_ variable in it. Now $eovowel is equal $_ ([e] or [o]). The current phoneme $_ is not printed (because it is [e] or [o]). The program reads a new line from the BLF file. The process of examining which phoneme follows the phoneme [e] or [o] in the BLF file repeats, because the $eovowel is again not empty.
15. If there are no more lines in the BLF file, it means that the $_ variable is empty and all the lines have been read. The BLF file is closed.
16. Then the program checks if the $eovowel is empty. The $eovowel variable is not empty if [e] or [o] is the last phoneme in the BLF file. Because there is no phoneme which follows (actually there is a pause in the BLF file, but the values of this vary last line may be only used for calculating the duration of the last phoneme; this problem is described in the section 7.3.3), the $eovowel has no chance to be printed while going through the "$eovowel

procedure" - when $_ is not empty, and $eovowel is not empty. Therefore, the value of the $eovowel variable must be checked after having read all the lines in the BLF files and closing the BLF file.

17. If the $eovowel variable is not empty, then the value of the $eovowel is printed. It is followed by printing the reconstructed final pause. Because it is not possible to count the duration of the final pause, the final pause gets the value of 200msec. The PHO file is closed.
18. If the $eovowel variable is empty, then the reconstructed final pause is printed into the PHO file and the PHO file is closed.

7.3.2 Which problems connected with the phoneme set conversion are not solved by the program?

The program does not deal with two Polish phonemes [c] and [J]. These phonemes are present in the BLF SAMPA annotation phoneme set, but are not included in the PL1 SAMPA phoneme set. These phonemes are reconstructed by the sequences of phonemes [kj] or [ki] for the phoneme [c] and by [gj] or [gi] for the phoneme [J]. The problem illustrates Table 8. The asterisk "*" at the brackets means that the phonemes in the brackets may or may not appear after the phonemes [c] and [J].

Table 8: The phonemes [c] and [J] from the BLF SAMPA annotation convention and their equivalents in the PL1 diphone database.

BLF SAMPA annotation labels		*PL1 SAMPA symbols*	
c	(j/i)*	k	j/i
J	(j/i)*	g	j/i

For the available annotation files, the occurrence of [c] and [J] was transcribed as [c] followed by [j] or [i] and [J] followed by [j] or [i], creating sequences of phonemes [cj] and [Jj]. Having these sequences of phonemes, it is easy to replace the phoneme [c] by the phoneme [k] and the phoneme [J] by the phoneme [g] without any additional sequential split. If the [c] or [J] were not followed by [j] or [i], then there would have to be introduced a process of splitting the phonemes [c] and [J] into sequences of [kj] or [ki] and [gj] or [gi], respectively. Then the duration of one phoneme would have to be split and one part of the value of the duration given to the phoneme [k] or [g] and the other part of the duration given to the phoneme [j] or [i].

It is hoped that all the other annotation files existing in the corpus transcribe the

occurrence of [c] and [J] in the same way, i.e. [c] or [J] followed by the phoneme [j] or [i]. If there are cases that [c] and [J] are not followed by the phoneme [j] or [i], then the described above sequential split will have to be introduced.

To illustrate the problem, different transcriptions of the word “kiedy” is presented in Table 9:

Table 9: Different transcriptions of the word "kiedy."

BLF notation	*BLF notation with the sequential split procedure needed*	*PHO notation*
c	c	k
j		j
e	e	e
d	d	d
y	y	I

7.3.3 Automatic duration calculation

Calculation of the phoneme durations in milliseconds from the time stamps with information on the sampling rate was not difficult. But it has to be stated that in the BLF files the time stamps are marked at the beginning of the phonemes in the BLF files, therefore the duration of the last segment cannot be measured. The formula for calculating the duration of phonemes is

$$(samplenumber_i - samplenumber_{i-1})/samplingrate.$$

In other words, the sample number from the following phoneme is subtracted from the sample number on which the program operates currently and the result of the subtraction is then divided by the sampling rate. This means that before printing a phoneme from a line, the program must read another line to calculate the phoneme's duration. To solve the problem, a variable $_ which stores the value of the previous phoneme is introduced. To calculate the duration, the value of the duration of the previous phoneme must be stored. This problem is also solved by introducing a variable called $sample_1. To be more explicit, the program:

1. stores values of the previous line,
2. reads a following line,
3. does the conversion of the phoneme from the previous line,

4. calculates the duration of the phoneme from the previous line making use of the values of the sample number on the current line,
5. checks if the phoneme from the previous line meets the printing conditions,
6. if the phoneme from the previous line meets the printing conditions, then this phoneme is printed. If it does not, then another line is read and the values from the penultimate and previous line are stored in other variables.

7.3.4 The monotone ACCS system

The first step in ACCS synthesis development was creating a monotone ACCS synthesis system. The inputs to that system were:

1. BLF annotation file,
2. Diphone database for the Polish language (the Polish Female Voice),
3. MBROLA engine.

Because the notation used in BLF format did not match PHO format, conversion algorithms were needed. Additionally, the phoneme set used in BLF annotation files differed from the set used by the Polish Female voice, which required another changes. Nevertheless, the final output of the system was a PHO file with monotone. The phonemes from the BLF annotation convention were converted into the phonemes used by the diphone database, which caused a small data loss. The original durations of the phonemes were preserved. The value of the pitch was added automatically and for all the phonemes was equal to 200Hz – a value characteristic for a female voice.

7.3.5 Praat pitch extraction

When the monotone ACCS synthesis system has been developed (see section above), the next step is to extract pitch from the original recordings. The extraction of pitch is one of the main objectives of the CCS system. Copying the phonemes, the durations of the phonemes from the annotation file and measuring the pitch values from the original recording of a human utterance allows to synthesise speech of the best possible quality.

To extract pitch from the recordings a Praat script called *max_pitch* was

implemented.[8] "This script goes through Sound and TextGrid files in a directory, opens each pair of Sound and TextGrid, calculates the pitch maximum of each labeled interval, and saves results to a text file" (Lennes, 2003).

The implementation of this script caused another problem and some modifications to the script were made.

The inputs to this Praat script are:

1. WAV files,
2. TextGrid annotation files.

The problem was that the available annotation files were in BLF format. That fact prompted the design of a new conversion algorithm. The approach taken was to convert the PHO files with monotone into TextGrid files. That choice was made, in contrast to converting BLF to TextGrid, because the script converting BLF phoneme set into the Polish Female Voice database already existed. Before the conversion algorithm was designed, the process of creating TextGrid files as well as integrating the pitch values into the PHO files was performed manually. (This manual integration is marked in Figure 14.)

Moreover, the automatic integration of the original pitch values into the PHO files was delayed, because the Praat pitch extraction file produces one TXT file with the pitch values of all the phonemes in the files in the directory. The output "pitchresults.txt" file contains following information:

1. filenames of the files in the directory,
2. labels,
3. maximum pitch values of the labelled intervals in Hz.

The *pitchresults* file for one file in a directory is shown in Figure 16. Because of the *pitchresults* format, the automatic integration of the original pitch values was left untouched at the beginning, creating a new transition step in developing the full ACCS synthesis system called *Semi-ACCS synthesis* in which data about pitch values were put into the PHO files manually.

8 The script was created by Mietta Lennes and is distributed under the GNU General Public License. I am grateful to the author of the Praat script for this freeware application.

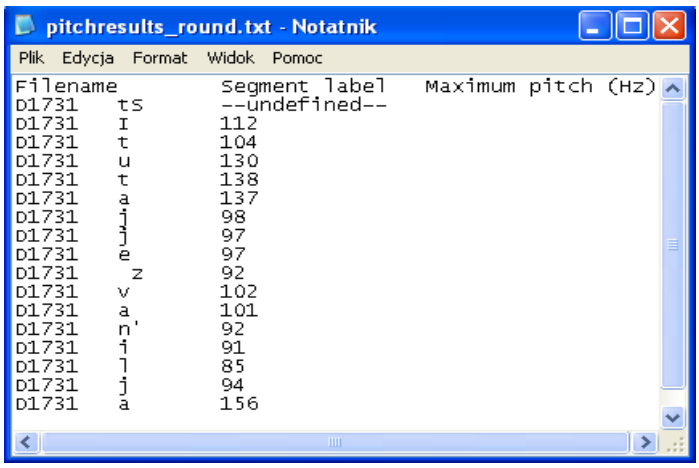

Figure 16: Pitchresults file generated by max_pitch Praat script.

The *max_pitch* Praat script, modified for the purpose of this study, is to be found in the Appendix C.

7.3.6 Inclusion of pitch values into MBROLA PHO file

The monotone pitch values from the first PHO file were replaced with the extracted pitch values using a Perl procedure. Although at the beginning the extracted pitch values with the use of *max_pitch* Praat script were put into the MBROLA PHO files manually in the procedure called *Semi-ACCS synthesis*, finally, an automatic inclusion of pitch values into MBROLA PHO files was developed. This procedure takes the *pitchresults* file generated by the modified *max_pitch* Praat script as an input. As described above the *pitchresults* file contains the names of all the WAV/TextGrid files (the WAV and TextGrid file names are identical) in a directory, labels and the maximum pitch for segments of these files. The problem was solved by dividing the *pitchresults* file into separate PITCH files in which there are only filenames, labels and pitch values for each file in a directory. Therefore, the PITCH files got the same length as TextGrid files. Similarly, PITCH files and MBROLA PHO files were almost equal in lenght. The difference was in the automatically generated first and last pauses in the PHO files. These pauses were removed by the Perl script and the PITCH and MBROLA PHO files were made the same length. Finally, the inclusion script takes:

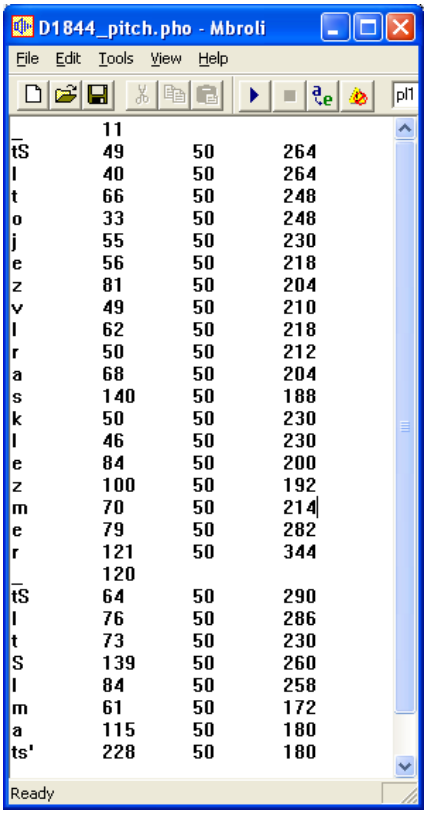

Figure 17: The automatically generated PHO file in ACCS synthesis procedure.

1. from MRBOLA PHO files with monotone:
    1. phonemes,
    2. duration of these phonemes,
    3. pitch position equal 50.
2. from PITCH files: pitch values.

Before printing new PHO files, a simple pitch emulation for the MBROLA female voice was introduced: the male pitch values are multiplied by two, because the original recordings are for a male voice and the diphone database is for a female voice. Furthermore, for phonemes where pitch value is undefined (the phonemes are voiceless), the script prints only phonemes and duration. The pitch pair (pitch position and pitch value) are left out. The automatically generated PHO file by the ACCS synthesis script is shown in Figure 17.

Figure 18 shows a waveform and a pitch contour of a human utterance *Michał podśmiewał się z kolegi, który dostał jedynkę ze sprawdzianu* (Michał laughed at a pupil who failed a test.) derived from the corpus. Below there is a waveform and a pitch contour of the same utterance, but synthesised with the ACCS synthesis procedure. Comparing both pitch contours there is not a big difference between them, which indicates that the prosody of the synthesised speech can be very human-like.

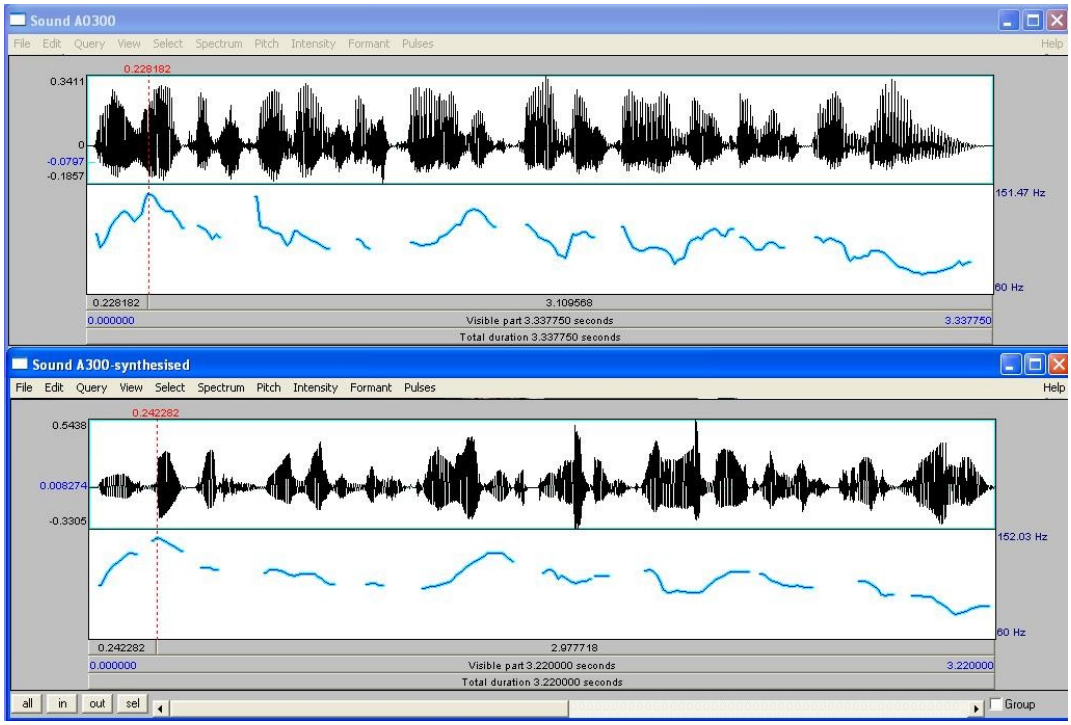

Figure 18: A waveform and a pitch contour of a human utterance "Michał podśmiewał się z kolegi, który dostał jedynkę ze sprawdzianu" and its synthesised equivalent using the ACCS synthesis.

7.3.7 BLF to TextGrid transformation software

BLF to TextGrid transformation software was developed because the *max_pitch* Praat script requires annotation files in TextGrid format. The step taken was to convert BLF notation into a generic XML notation, for which a library of functions, including TASX to Praat, already exist. This script depends on an additional Java runtime environment and the Saxon XML engine, and therefore the conversion process is more complicated. In the final version of the ACCS synthesis system this software was not used. The reason for taking that step was that algorithms converting the set of phonemes used in annotation files into the set of phonemes used by the Polish Female Voice had already been developed for creating PHO files. Therefore, in the ACCS speech synthesis system TextGrid files are made on the basis of PHO files, not the original BLF files.

CHAPTER 8 Evaluation

8.1 Overview of speech synthesis evaluation

Humans are exposed to speech throughout their lifetime, and they are very sensitive to small changes in speech quality. Furthermore, from the speech signal humans do not only infer information conveyed through the words themselves, but are also able to say a lot about the non-verbal elements of speech which modify meaning and convey emotions. A human listener easily detects information about the speaker's age, gender, accent and many other characteristics (Lampert 2004:3).

Although the first known attempt to create a talking machine was in 1003AD when Gerbert d'Aurillac was supposed to have built a bronze head, or to have acquired it from the Nine Unknown Men, which would answer his questions with "yes" or "no"[9], the quality of synthetic speech is still below of that produced by human beings in many aspects. One big difference between human speech and synthetic speech is that the latter is much more difficult to understand by listeners who are unaccustomed to synthetic speech. People who are not used to synthesised speech have to put more effort in order to understand the message conveyed in human-like synthetic speech signal. Techniques are being developed to produce "natural-sounding" synthetic speech, i.e. synthetic speech which is as easy to understand as human speech (Hawkins et al. 2000: 13/1).

In order to improve speech synthesis systems and to be able to compare different speech synthesisers, a thorough evaluation of different kinds of TTS systems must be conducted. Evaluating a speech synthesis system is not an easy task, because there are many aspects of synthetic speech and the systems themselves which should be

9 Wikipedia, the free encyclopedia; consulted 2007-02-22.

investigated. Nevertheless, the most popular and probably unavoidable method of synthetic speech evaluation is to have humans listen to the synthetic speech and respond to specific questions or make subjective judgements (Lampert 2004: 3). But this method of evaluation is time-consuming and expensive, therefore scientists work on creating new automatic methods which would not require humans take part in the evaluation procedure. All the methods of speech output assessment and criteria which are taken into account when evaluating speech will be dealt with in this chapter.

8.1.1 Taxonomy of methods of TTS evaluation

TTS system evaluation is a multi-dimensional task and there are different approaches to it. Many different types of evaluation procedures were developed and each has different evaluation goals and requirements. Several attempts have been made to create a taxonomy for the types of speech output assessment (Van Bezooijen & Pols 1990, Jekosch & Pols 1994, Goldstein 1995). Based on these taxonomies the Expert Advisory Group on Language Engineering Standards (EAGLES) proposed their own taxonomy of speech output assessment. The taxonomy is shown in Figure 19 and described below (Gibbon et al. 1997).

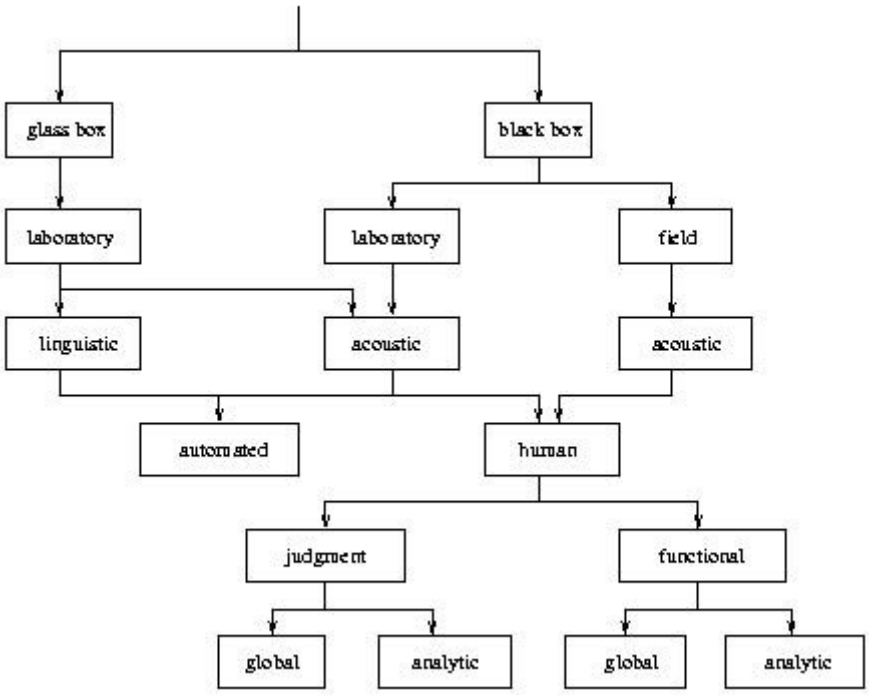

Figure 19: Relationships among methods of speech output evaluation involved in a taxonomy. From Gibbon, Moore and Winski (1997: 486).

8.1.1.1 Glass box vs. black box

The first level of the taxonomy proposed by EAGLES makes distinction between glass box and black box types of assessment methods. Glass box assessment methods are used for diagnostic purposes in order to improve the speech output system in question. This method focuses on testing specific components of a system. Black box assessment methods, on the other hand, focus on the performance of a system as a whole. Any internal structure must be known. The system is considered as a black box which accepts text and outputs speech. Black box assessment methods are used for comparative testing, for example when comparing different systems or in order to trace the improvement of one system which has been made over time.

Glass box assessment methods are useful to researchers and TTS systems developers who intend to improve the systems and find bugs in the program, but the glass box methods are not of much interest to end users. On the other hand, black box assessment methods can be useful for users, researcher and developers, because they give assessment of the whole system, not only specific components.

Following this distinction between the glass box and the black box, Pallett and Fourcin (1996) proposed a strict terminological division between "assessment" and "evaluation":

1. **Assessment** is the process of system appraisal which leads to global, overall performance. Assessment is related conceptually to black box methods (or: performance evaluation) in which the detailed mechanisms of processing are not considered.

2. **Evaluation** involves the analytic description of system performance in terms of certain factors, it is concerned with detailed measurement. Evaluation is conceptually related to the glass box approach (or: diagnostic evaluation), in which the objective is, for example, to gain a greater understanding of the system performance from the use of precision diagnostic techniques based on special purpose phonetic databases (Pallett & Fourcin 1996: 404).

8.1.1.2 Laboratory vs. field testing

The second level of the taxonomy differentiates between laboratory and field testing. Laboratory tests are used in the glass box assessment, whereas the black box

assessment methods can make use of both laboratory and field tests.

Laboratory tests characterise precision and control over unpredictable factors while testing a TTS system. In laboratory tests, system components or specific applications of a system are tested under controlled conditions. On the other hand, in the field, the actual performance of a system with end users in real life is tested.

8.1.1.3 Linguistic vs. acoustic

The next level of the taxonomy distinguishes between methods of assessing the linguistic and acoustic modules in a TTS synthesis system. When assessing a TTS system, the performance of converting a text to a phonemic transcription is tested at the linguistic level. Then the phonemic transcription code is converted into speech output and this speech output is tested at the acoustic level. Testing transcription may be conducted manually or automatically, while testing audio requires humans listen to the sound.

8.1.1.4 Human vs. automated

The distinction between human evaluation and automatic evaluation is the next level of the taxonomy by Gibbon, Moore and Winski. Black box testing tends to require humans listen to the synthetic speech. However, the tests requiring humans are expensive, time-consuming, and furthermore, a lot of subjects have to take part in the test to make the test result significant statistically. Humans tend to be inconsistent in their judgements or task performance, therefore the results are somewhat noisy. Nevertheless, the speech synthesis systems are developed to be used by humans, so it is obvious that a TTS system has to be tested with end users, i.e. human beings.

On the other hand, researchers try to develop tests which would not involve human evaluation. Automatic assessment is very desirable, because any automatisation eases the task that would have to be performed by a human. The results of automatic assessment are not noisy, because the assessment system gives answers consistently according to some rules. For example, automatic assessment may test symbolic phonemic transcription component output against a pronunciation dictionary (Lampert 2004: 6).

8.1.1.5 Judgement vs. functional testing

Judgement testing (or: opinion testing) is by nature subjective. In a judgement test a group of listeners is asked to judge the performance of a speech output system along a number of rating scales. For example, listeners may be asked to judge the naturalness of synthetic speech on a scale from 0 to 20.

On the other hand, functional testing assesses the actual performance of TTS system in a communicative situation. For example, listeners may be asked to identify synthesised sounds. Speech intelligibility and comprehension tests belong to the group of functional tests.

8.1.1.6 Global vs. analytic assessment

The last level of the taxonomy makes distinction between global and analytic assessment. Judgement tests usually include tests where the subjects rate such global aspects of synthetic speech as "overall quality", "naturalness" and "acceptability".

In the analytic testing, the quality of specific aspects of a speech output system is assessed. Namely, listeners are requested to pay particular attention to selected aspects of the speech output.

8.1.2 Criteria for TTS evaluation

The methods of TTS evaluation described above aim at evaluating different aspects of a speech output system. The criteria which may be tested in a TTS system are listed below (Gibbon & Moore & Winski 1997):

1. NLP module:
    1. Glass box approach:
        1. Linguistic aspects:
            1. Text preprocessing:
                1. abbreviations, e.g. "i.e." → that is,
                2. acronyms, e.g. "UN" → you en,
                3. numbers, e.g. "124" → one hundred and twenty four, "1:24" → twenty four minutes past one,
                4. special symbols, e.g. "#1" → number one, "£1.50" → one pound fifty.

2. Parser:
    1. Sentence demarcation.
    2. Phrase demarcation. Phrases can be characterised further in terms of:
        1. Labelling errors,
        2. Expansion errors.
    3. Syntactic parsing – development and evaluation of this component does not belong to the domain of speech output systems. Syntactic parsing is much more of the interest of a language engineering, needed in automatic translation systems, grammar checking, and the like.
3. Grapheme-to-phoneme conversion:
    1. phonemic representation of words in isolation,
    2. phonemic representation of words in context, due to:
        1. assimilation to adjacent words, e.g. *I have **to** go* vs. *I have **two** goals*,
        2. heterophonous homographs: *I **lead*** vs. *made of **lead.***
4. Word stress:
    1. the coverage of stress rule module.
5. Sentence accent:
    1. accent placement rules.
6. Morphological decomposition – the accuracy of breaking long and complex words into smaller basic words and affixes (i.e. morphemes).

2. Prosody:
    1. Judgement tests of prosody:
        1. Quality of synthetic speech melody – intonation-by-rule module:
            1. Word level:
                1. consonant duration rules,
                2. vowel duration rules,
                3. stressed syllable.
            2. Utterance/Sentence level:

1. boundary marking rules:
    1. pitch rules,
    2. preboundary lengthening rules.

2. Functional tests of prosody:
    1. Disambiguation of:
        1. minimal stress pairs,
        2. word boundaries,
        3. constituent structure,
        4. focus distribution.
    2. synthesis of emotion – emotion conveyed in synthetic speech:
        1. emotion-by-rule module.

2. DSP module:
    1. Black box approach:
        1. Functional laboratory tests:
            1. Comprehensibility of speech output:
                1. intelligibility at a paragraph level - short paragraphs presented, preferably with human produced versions as a topline control,
                2. intelligibility at sentence level:
                    1. Psychoacoustic tests:
                        1. indexing of cognitive workload,
                        2. assessment of speed of comprehension.
    2. Glass box approach:
        1. Acoustic aspects:
            1. Segments:
                1. Segmental tests at word level:
                    1. Functional segmental tests at word level:
                        1. Correct phoneme identification:
                            1. consonants, e.g. in word initial position, in word medial position, in word final position.
                            2. vowels.
                        2. Correct cluster identification:
                            1. consonant clusters, e.g. tautosyllabic, heterosyllabic.

2. vowel clusters.
3. Intelligibility of unstressed syllables,
4. Intelligibility of de-accented or cliticised words,
5. Intelligibility of words in sentences,
6. Intelligibility of polysyllabic words,
7. Insertions and deletions.

2. Judgement tests at the word level:
1. Assessment of naturalness, intelligibility, and pleasantness of consonant clusters in both initial and final position.

3. Segmental tests at sentence level:
1. Functional segmental tests at sentence level:
1. segmental intelligibility – identification of keywords:
1. nouns,
2. verbs.

2. Judgement tests at a sentence level are used in black box approach to evaluate overall output quality.

3. Voice characteristics:
1. correct encoding of speaker characteristics, e.g. sex, age, regional background in terms of, for example:
1. acoustic aspects, e.g. mean pitch level, mean loudness, mean tempo, harshness, creak, whisper,
2. articulatory aspects, e.g. tongue body orientation,
3. functional aspects, e.g. dialect, accent.

2. voice pleasantness - unpleasant voice quality may be used for systems reading popular scientific books and newspapers, but unfits systems reading poetry and novels.
3. adequacy of voice quality for the purpose.

3. Holistic tests - Overall system performance:
1. Black box approach:
1. Judgement laboratory tests – subjects can indicate their subjective impression of global quality aspects of synthetic output by means of rating scales.

1. Overall speech quality:
    1. *Intelligibility* (How much identifiable is the message?),
    2. *Naturalness* (To what extent does the synthetic speech sound like being produced by a human speaker?),
    3. *Acceptability* (To what extent is the user satisfy with the communicative situation?).
2. Listening effort,
3. Comprehension problems,
4. Precision of articulation,
5. Accuracy of pronunciation,
6. Pleasantness of voice,
7. Adequacy of word stress,
8. Appropriateness of tempo,
9. Liveliness,
10. Fluency.

3. Field testing – gives the possibility to see how a speech output system functions in real life. Criteria taken into account when testing a TTS system in a field may be:
    1. subjects' attitudes towards the technology,
    2. DLP module:
        1. intelligibility of CVC-words,
        2. speech quality,
    3. subjects' proficiency in using a TTS application, e.g. is it easy for the users to find an article in a newspaper in an electronic newspaper for the visually handicapped?
    4. actual performance of the system in the real life:
        1. Are people able to understand information given by, e.g. electronic information service?
        2. Is quality of synthetic speech when listening to through a telephone line as intelligible as when listened to through a good quality headphones?
        3. Are people able to understand synthetic speech while doing something else, i.e. when their eyes and hands are

busy with some secondary task?

4. How a TTS system performs in noise?

Having listed the criteria which may be evaluated in a TTS system, it has to be said that different comprehension tests may make use of different types of questions and different answering modes. Two types of question which may be asked to the subjects are:

1. open questions,
2. closed (multiple choice) questions.

The first type of question is more sensitive. It does not impose or suggest an answer to the subject. On the contrary, closed questions give a variety of answers and the subject may always give a good answer just by chance.

Additionally, different answering modes may be adapted by the comprehension tests:

1. off-line tests - subjects answer questions after the texts have been presented,
2. on-line tests – psycholinguistically oriented, require instantaneous reactions to the auditory material being presented. These tests aim at gaining insight into the cognitive processes underlying comprehension: what is the difference between processing human speech and synthetic speech?

In the following section, examples of different kinds of tests are presented:

1. Functional segmental tests aimed at testing DSP module:
    1. The Diagnostic Rhyme Test (DTR) → identification of initial consonants in meaningful CVC-words. E.g. The subject indicates if the stimulus was *dune* or *tune*.
    2. The Modified Rhyme Test (MRT) → identification of the initial and final consonants in meaningful CVC-words (identification of the initial and the final consonant never tested simultaneously). E.g. The subject indicates what was the final consonant in a contrastive syllable coda series such as *peas, peak, peal, peace, peach*, and *peat*.
    3. The Standard Segmental (SAM) Test → identification of consonants in word initial, word medial and word final position in meaningless and

(sometimes by chance) meaningful CV, VC, VCV stimuli, e.g. *pa, ap, apa*, *ki, ik*, and *iki*.

4. The Diphone Test → identification of consonants and vowels in all permissible (pronounceable) CVC, CVV, VCV, VCCV sequences in a given language.
5. The Cluster Identification (CLID) Test → identification of single consonants, consonant clusters (tautosyllabic and heterosyllabic), single vowels, vowel clusters in both meaningful and meaningless items which can be generated for a given language according to its phonotactic rules.
6. The Ballcore Test → identification of tautosyllabic consonant clusters and single clusters in a fixed set of CVC-stimuli, e.g. *frimp* and *friend* or *glurch* and *parch*.
7. The (Modified) Minimal Pairs Intelligibility Test and the Diagnostic Pairs Sentence Intelligibility Evaluation (DPSIE) Test → intelligibility of initial consonant, vowels, tautosyllabic (same syllable) as well as heterosyllabic (different syllable) consonant clusters, unstressed syllables, de-accented or cliticised words, words in sentences, polysyllabic words, insertions and deletions in a fixed set of 265 sentence pairs containing one contrast, e.g. *The horrid courts scorch a revolution.* vs. *The horrid courts score a revolution*.

2. Functional segmental tests at sentence level:
    1. Harvard Psychoacoustic Sentences → identification of keywords (nouns and verbs) in a fixed set of 100 semantically and syntactically "normal" Harvard Psychoacoustic Sentences. E.g. *Add salt before you fry the egg.*
    2. Haskins Syntactic Sentences → identification of keywords in the fixed set of 100 Haskins Syntactic Sentences. E.g. *The old farm cost the blood.*
    3. Semantically Unpredictable Sentences (SUS) → identification of keywords in a fixed set of five syntactic structures which are common in most Western European languages, such as "Subject-Verb-Adverbial" E.g. *The table walked through the blue truth.*

4. Judgement laboratory tests in which subjective overall impression of speech quality is tested are:
    1. *Paired comparison*, in which subjects indicate which of two synthesisers produces synthetic speech which is more comprehensible.
    2. *Magnitude estimation*, where subjects express their impression of comprehensibility of the synthetic speech by assigning a value, or by drawing a line of a length which is equal to the magnitude of their impression of comprehensibility.
    3. *Categorical estimation,* in which subjects rate different synthesisers, for instance, along a 10-point rating scale which runs from 1: extremely incomprehensible to 10: extremely comprehensible.
3. Psychoacoustic tests:
    1. The *word monitoring task* indexes the cognitive workload. In this test subjects press a button as soon as they hear a prespecified word.
    2. The *sentence-by-sentence listening task* assesses the speed of comprehension. The subjects' task is to press a button whenever they are ready to listen to the next utterance (comprehension of the sentences is checked afterwards but is not part of the test itself).
    3. The *sentence verification test* also assesses the speech of comprehension. In this test subjects are asked to decide whether short sentences are true statements or not, e.g. *Mud is dirty* and *Rockets move slowly*.

8.1.3 Relevant criteria and methods for ACCS evaluation

Some of the criteria for TTS evaluation can serve the purpose of ACCS evaluation. Although there are some components of the ACCS system which cannot be tested, e.g. the text preprocessing and grapheme-to-phoneme conversion modules, because the ACCS synthesis system does not accept raw text as an input, nevertheless, there are many other criteria which may be tested in the ACCS system, and which are common to all speech synthesis system evaluations. The criteria and the methods which can be tested in the ACCS synthesis system are listed below:

1. DSP module, black box approach:

1. Laboratory testing:
    1. Functional laboratory tests:
        1. Comprehensibility of speech output:
            1. Intelligibility at sentence level:
                1. The *word monitoring task* test.
        2. Judgement laboratory tests:
            1. Overall speech quality:
                1. Intelligibility,
                2. Naturalness.
            2. Listening effort,
            3. Comprehension problems,
            4. Precision of articulation,
            5. Accuracy of pronunciation,
            6. Pleasantness of voice,
            7. Adequacy of word stress,
            8. Appropriateness of tempo,
            9. Liveliness,
            10. Fluency.
2. DSP module, glass box:
    1. Acoustic aspects:
        1. Segments:
            1. Segmental tests at word level:
                1. Functional segmental tests at word level:
                    1. Correct phoneme identification: consonants and vowels.
                    2. Correct clusters identification: consonants and vowels.
            2. Segmental tests at sentence level:
                1. Functional tests:
                    1. Segmental intelligibility – identification of keywords: nouns and verbs.
                    2. Intelligibility of isolated sentences: meaningful and meaningless sentences.
3. Technical aspects – the performance of the ACCS system modules:
    1. Objective evaluation of actual pitch/duration in re-synthesised speech:

1. BLF to PHO conversion – a component of the Perl script,
2. *max_pitch* performance:
    1. Judgement tests of prosody:
        1. quality of synthetic speech melody, e.g. in statements, questions, multiple sentences and exclamations.
2. Perl script performance,
3. Correctness of annotations.

8.2 Evaluation of the system

In the following section, selected evaluation tests are presented. These tests serve to assess the ACCS system, and at the same time, check the quality of the annotated speech corpus on which the ACCS is based. Two types of evaluation were performed: diagnostic tests of the ACCS system and speech output assessment tests.

Firstly, diagnostic tests evaluating the performance of the ACCS program were carried out. Those evaluation tests investigated the performance the Perl script with its subcomponents, and the pitch extraction component, the *max_pitch* Praat script. Moreover, a preliminary evaluation of the correctness of annotations (the input to the CCS system) was done.

Secondly, a preliminary test of comprehensibility of the synthetic speech was carried out on one subject. The satisfactory results of that test were the basis for designing and performing speech output assessment tests. Those tests tested not only the intelligibility and naturalness of speech, but also served to assess the quality of the annotations. The speech output assessment tests were carried out on nineteen native speakers of Polish and two foreigners with good command of Polish.

The diagnostic tests and speech output assessment tests and their results are presented below.

8.2.1 Diagnostic evaluation

The program was tested on approximately 880 BLF and 880 WAV files. The testing material was divided into smaller groups. At the beginning, the groups consisted of 40 BLF and 40 WAV files. Later, the material was grouped into folders with up to 50 BLF files and 50 WAV files in one folder and one folder consisted of 100 BLF files

and 100 WAV files.

8.2.1.1 The incomplete performance of the program

In the course of testing, it was discovered that the program created correct PHO files for all the files in a directory, i.e. 40, 50 or 100 PHO files, but failed to play all the files. Up to 45 synthesised PHO files were played one by one at one runtime of the program. Then the program terminated correctly (removing all the files that were automatically created for the synthesis.) However, the program usually terminated after playing a part of the files in the directory.

Synthesis in the folders with 40 BLF and 40 WAV files was correct for all the files and in some folders (five folders) all the newly created PHO files were played correctly all at once. However, in none of the folders with 50 BLF and 50 WAV files in which new 50 PHO files were correctly created, the PHO files were played all one by one. The *phoplayer* stopped playing the files after playing approximately half of the PHO files in a directory, i.e. 25 PHO files (35 PHO files – maximum, 15 PHO files – minimum). In the folder with 100 BLF and 100 WAV files the synthesis was carried out correctly for all the files, i.e. 100 new PHO files were correctly created. But program stopped playing the PHO files after having played 15 files.

The reason for the incomplete program's performance is not known yet. During the testing procedure a few steps were taken:

Action: Result:	The BLF file at which the program stopped running was removed. The program played the BLF files which preceded and followed the removed BLF file at the second run, but stopped anyway, before playing all the files if there were too many files in the directory, i.e. over 20 files.
Action: Result:	The BLF file at which the program stopped was removed, together with one preceding and one following BLF file. When the program was run again, it played the BLF files which preceded and followed the removed BLF files, but stopped anyway, before playing all the files if there were too many files in the directory, i.e. over 20 files.

Action:	All the files which preceded the BLF file at which the program terminated were removed.
Result:	The BLF file at which the program stopped running was played together with all the following BLF files if there were not too many files in the directory, i.e. over 20 files.
Action:	All the files which preceded the BLF file which was before the BLF file at which the program stopped were removed. In other words, the BLF file which preceded the BLF file at which the program terminated was saved, but all the other preceding BLF files were removed from the directory.
Result:	The first BLF file in the directory and the BLF file at which the program stopped running were played correctly at the second run of the program. Also all the following BLF files were played if there were not too many files in the directory, i.e. over 20 files.

The cause of playing only some part of the BLF files in the directory is not known and must be investigated in the future. It is expected that the problem itself is not caused by the program (all the PHO files in a given directory are created correctly), but the problem lies in the *phoplayer* which is called from the main synthesis program to play the PHO files.

For the current use, the problem with the *phoplayer* causes nuisances, but does not indicate errors in the program. The program produces correct PHO files, and the only disadvantage caused by the unreliable *phoplayer* is that the user has to play each PHO file manually.

8.2.1.2 BLF and WAV files in the directory

The structure of the program does not allow it to run the program in a folder in which there are BLF files without corresponding WAV files. It is possible to have more WAV files in a directory on which the program is run, but it is not allowed to have a BLF file without the corresponding recording. An automatic check for this has not been included in the program so far.

8.2.1.3 Errors in the annotation files

The program could not run correctly when it came across annotation files with errors.

In the BLF files two kinds of errors appeared:

1. the final pause was incorrectly set,
2. wrong phoneme labels were used.

Firstly, it was discovered that in some BLF files the final pause had been set incorrectly by the automatic annotator. A BLF file with the incorrectly set final pause is shown in Table 10. The first column of Table 10 shows the sample numbers from a BLF file; the example was derived from the corpus. It is easy to see that the sample number of the last segment, the final pause, is smaller that the previous sample numbers. It means that this segment should appear before the segments that are present in the table, somewhere else in the annotation file. Unfortunately, the incorrectly set final pauses by the automatic annotator were not always found in the process of the manual correction of the annotations, therefore such an error occurs in some BLF files. The program cannot run if it comes across a file with such an error. Fortunately, it is easy to find automatically the file with the incorrectly set final pause and remove it or correct the annotations by hand.

Table 10: Fragment of BLF file with incorrectly set final pause; the example is derived from the corpus.

Sample number (16 kHz rate)	*Segmental labels*	*Prosodic labels*
133292	#t^S	
135002	"y	
135740	.l	
136800	i	
137760	#g	5,.
138763	"a	
140640	.m	
141703	e	
143139	t	
66100	#$p	

Secondly, in some of the BLF files wrong phoneme labels were used. The mistakes were made during the process of manual annotation. This could have been expected, since to err is human. Some of the incorrect phoneme labels were corrected automatically, but there were a number of errors which had to be corrected manually, since the character of the mistake was not known, i.e. it was not known which phoneme label the manual annotator intended to insert.

Although in that test only errors on the phoneme level were taken into account, in the future work on Parametric CCS also the prosodic labels will be checked to validate the annotations.

8.2.2 Speech output assessment – naturalness & comprehensibility

8.2.2.1 What is speech quality?

Speech quality is a a multi-dimensional term. Jekosch (2005: 6) defines the term speech quality as follows:

> ***Speech quality***
>
> *The result of assessing all the recognized and nameable features and feature values of a speech sample under examination, in terms of its suitability to fulfil the expectations of all the recognized and nameable features and feature values of individual expectations and/or social demands and/or demands.*

8.2.2.2 Speech quality tests

The preliminary evaluation of the synthetic speech was the basis for designing speech quality tests. The preliminary test was carried out on one listener, who made overall judgements of the speech quality. That person made the manual corrections of the material, so she knew it, and was, to some extent, used to synthetic speech. The subject was presented with 880 synthesised sentences from the corpus (Base A, Base B and a part of Base C). In the course of the preliminary evaluation of the automatically close copied synthetic speech the fallowing was observed:

1. The synthesised speech was understandable after having listened to it only once.
2. The prosody of the synthesised speech was very good.
3. Some interruptions in the speech signal occurred, but they did not make the synthetic speech difficult to understand.

These promising results fed into designing and carrying out speech quality tests presented below. The stimuli for these tests were based on the annotation files from the corpus. The annotations were correct: they did not include the errors mentioned

before, i.e. incorrectly set final pause or usage of wrong phoneme labels.

The speech output assessment tests are described below. The test materials are to be found in Appendix D. The answer sheets are included in Appendix E.

8.2.2.2.1 Test 1, sentence and word recognition – functional testing of intelligibility of speech from glass the box approach

Method:	Meaningful and meaningless synthesised sentences from the corpus (Base A and Base B) are presented to the subjects. The subjects write down what they hear in the answer sheet. The set of meaningless sentences is used to eliminate the influence of the top-down processing (Clark & Yallop 1995: 312, Ryalls 1996: 94).
Material:	10 meaningful and 10 meaningless synthesised sentences with the female voice.
Instructions:	In a moment you will hear 20 sentences. Your task is to write down the sentences. After each sentence, there is a few-second pause. This is the time for you to write down the sentence.

8.2.2.2.2 Test 2, subjective sentence quality test – judgement testing of speech quality from the black box approach

Method:	The subjects are asked to evaluate the quality of isolated long (multiple) sentences from the corpus (Base E) at 5 level scale: Excellent – Good – Fair – Poor – Bad.
Material:	10 different compound sentences have been chosen from the corpus. 15 compound sentences were synthesised using the female diphone database: 1. 10 sentences have a value of pitch adapted for a female voice. This voice is called *pseudo-female*. 2. 5 sentences from the set have the original pitch values extracted from the recordings of a male speaker. This voice is called *pseudo-male*. Additionally, 5 sentences (sentences which were not synthesised with the pseudo-male voice) uttered by a human professional speaker are

	added. These are the original recordings from the corpus. Altogether, 20 sentences are used in the test, but only 10 different sentences. All the sentences are played in a random order. The results of the set of 5 sentences synthesised with the pseudo-female voice and the results of the set of 5 the same sentences synthesised with the pseudo-male voice are to be compared. Similarly, the results of the set of 5 sentences synthesised with the pseudo-female voice and the results of the set of 5 the same sentences uttered by a male speaker are going to be compared.
Instructions:	In a moment you will hear 20 long sentences. Your task is to evaluate the quality of the speech. After each sentence, there is a few-second pause. This is the time for you to decide which of the five grades you would give to the utterance: Excellent – Good – Fair – Poor – Bad.

8.2.2.2.3 Test 3, isolated word intonation test – judgement testing of prosody from the glass box approach

Method:	Isolated words are presented to the subjects. The subject decides whether the words would appear at the end of a statement or at the end of a question based on the intonation of the word.
Material:	A set of 20 words cut out of the whole sentences from the corpus (Base D). These words originally appeared at the end of a statement, a question, an exclamation and at the end of a continuation phrase, e.g. To nie są banały tylko błogosławieństwo. Czy dał błogosławieństwo? Przecież tutaj jest napisane błogosławieństwo! Wyraz błogosławieństwo - w języku polskim - niewiele znaczy. The words from the exclamations and continuation phrases are distractors, and are not the subject of the study. But it is assumed that exclamation-words will be recognised as statement-words because of their falling intonation, and continuation-phrase-words will be recognised as question-words because of their rising intonation. The words from the set of words are presented in a random order.

Instructions:	In a moment you will hear 20 words. After each word there is a few-second pause. Your task is to assess if the melody of the word suggests that this word would appear at the end of a statement or at the end of a question. For the items for which you cannot decide, mark "Don't know."

8.2.2.3 Results and discussion

The three speech quality tests were administered to nineteen Polish subjects and two foreigners. The youngest Polish subject was eight years old, the oldest was fifty-five years old. The foreigners' ages were twenty-two and twenty-five years old. The subjects took the tests separately. The stimuli were played once or twice to the subjects, depending on the subject's needs. Each testing session lasted from 30min to 45min, depending on the subjects personal needs.

The tests had field character, they were administered in different, but silent places, both indoors and outdoors. To the tests a laptop and standard built-in loudspeakers were used. This kind of equipment was chosen, because it eased field work.

8.2.2.3.1 Results: Test 1

The test results for Test 1 are presented in Table 11. Detailed test results for all the subjects are shown in Appendix F.

Table 11: Results for Test 1 – average correctly recognised units in all sentences. N stands for the number of subjects.

	N	*Sentences (absolute)*	*Sentences (percent)*	*Words (absolute)*	*Words (percent)*
Polish male	8	13,63	68%	111,00	88%
Polish female	11	14,36	72%	115,82	92%
Polish overall	19	14,05	70%	113,79	90%
Foreigners	2	2,00	10%	61,00	48%

Table 11 shows the test results for the Polish male and female subjects and their overall. Moreover, the Table presents the results of the foreigners who took the test. The results are presented in both, the average number of recognised sentences and words, as well as their percentages. The results are as follows:

1. Sentence:

1. Male subjects: The male subjects correctly recognised 13,63 (68%) sentences from the set of 20 sentences.
2. Female subjects: 14,46 (72%) sentences were recognised correctly.
3. Overall: The average recognition of sentences by Polish subjects was 14,05 sentences (70%).
4. Foreigners: Foreign subjects correctly recognised 2,00 (10%) sentences.

2. Words:
    1. Male subjects: The male subjects correctly recognised 111,00 (88%) words from the set of 20 sentences including 126 words.
    2. Female subjects: 115,82 words (92%) were recognised correctly.
    3. Overall: The average recognition of words by Polish subjects was 113,79 (90%) words.
    4. Foreigners: Foreign subjects correctly recognised 61,00 (48%) words.

Very good results at the word level for the Polish subjects show that the automatically close copied speech is highly intelligible. The results at the sentence level are worse, but also promising if it is taken into account that 10 sentences in the test were semantically predictable and 10 sentences were semantically unpredictable. The comparison of the separate results for the semantically predictable and unpredictable sentences are shown in Table 12; only the results for the Polish listeners are investigated.

The results show that 83,30% of the semantically predictable sentences were recognised correctly which makes up the recognition of 96,28% words in these sentences. The results of the unpredictable sentences are much worse on the sentence level – that is 55,30% of correctly recognised sentences. But the recognition of separate words in these sentences is also very high – 81,53% words.

The comparison of the results of the meaningful sentences with the results of the meaningless sentences shows that when the top-down component is eliminated from the speech perception process, the recognition of single words and whole sentences is much worse (sentences – 28,00% worse, words – 14,75% worse). Nevertheless, the recognition of words in the semantically unpredictable sentences is still very high,

which means that the synthesis of the stimuli was very good.

Table 12: The comparison of the results for the semantically predictable and unpredictable sentences, Polish subjects only. N stands for the number of items.

	N	*Predictable*	*N*	*Unpredictable*
Sentences	10	83,30%	10	55,30%
Words	75	96,28%	51	81,53%

8.2.2.3.2 Results: Test 2

The test results for Test 2 are presented in Table 13. Detailed test results are shown in Appendix F.

Table 13: Test results for Test 2. N is the number of subjects, MOS score/5 stands for Mean Opinion Score out of 5, STDV is the standard deviation, Max:Min are the maximal and minimal scores given by the population.

	N	*Original*			*Pseudo-female*			*Pseudo-male*		
		MOS score/5	*STDV*	*Max: Min*	*MOS score/5*	*STDV*	*Max: Min*	*MOS score/5*	*STDV*	*Max: Min*
Polish male	8	4,70	0,52	5:3	3,03	0,83	4:1	2,45	0,75	4:1
Polish female	11	4,69	0,50	5:3	2,50	0,87	4:1	2,18	0,89	4:1
Polish overall	19	4,69	0,51	5:3	2,73	0,89	4:1	2,29	0,85	4:1
Foreigners	2	4,50	0,71	5:3	2,30	0,81	4:1	1,30	0,48	2:1

Table 13 shows the test results for Polish male and female subjects, their overall and results for the foreign subjects. In the tests different voices received the following average scores:

1. Original voice:
    1. Male subjects graded the original voice with 4,70.
    2. Female subjects graded the original voice with 4,69.
    3. In the overall score the original voice received 4,69.
    4. Foreigners graded the original voice with 4,50.
2. Pseudo-female voice:
    1. Male subjects graded the pseudo-female voice with 3,03.
    2. Female subjects graded the pseudo-female voice with 2,50.
    3. In the overall score the pseudo-female voice received 2,73.
    4. Foreigners graded the pseudo-female voice with 2,30.
3. Pseudo-male voice:
    1. Male subjects graded the pseudo-male voice with 2,45.

2. Female subjects graded the pseudo-male voice with 2,18.
3. In the overall score the pseudo-male voice received 2,29.
4. Foreigners graded the pseudo-male voice with 1,30.

In the test the best scores received the original voice. Both Poles and foreigners graded it highly. The range of the grades given to the original voice was the same in both groups with the minimal grade 3 and the maximal 5.

The synthetic pseudo-female voice received much worse scores from Poles and foreigners. It was graded approximately two point less (in the five-point rating scale) than the original voice. This result did not vary significantly from the score which received the synthetic pseudo-male voice in the evaluation by Poles. The pseudo-male voice scored just almost half the point less (precisely 0,44 point less) than the pseudo-female voice. However, foreigners graded the pseudo-male voice worse of one point.

Both groups assigned varied grades to the synthetic voices which ranged from 1 to 4. However, foreigners were very stable in their evaluations of the pseudo-male voice. The minimal grade given was 1 and the maximal was 2.

The results show that the subjects graded the human voice much better than the synthetic voices. But it has to be underlined that the synthetic voice was confronted with recordings of a professional speaker recorded in a professional studio. As expected, the pseudo-female voice was evaluated better than the pseudo-male voice, although the scores were not very different. This suggests that the pseudo-male voice seemed as natural and intelligible as the pseudo-female voice.

Much worse results of the pseudo-female and pseudo-male voices may suggest that the intelligibility and articulation in the synthetic signal was not very good. The problem might lie in the annotations or in the diphone database itself.

It is also taken into account that the subjects being aware of evaluating a machine, i.e. synthetic voice, graded it with lower grades just “in case”. However, after the testing procedure the subjects were asked informal questions about what they heard and it turned out that they did not realise they were exposed to synthetic speech. They believed that they were evaluating articulation of human speakers. These informal observations are very promising.

Additionally, the results for the Polish male and the female subjects were compared to see if the results of the two groups show statistically significant differences. For this purpose two methods were used. One method is rather informal comparing the dispersion ranges of two sets, the other is the T-test or so called student test.

The first method compares the dispersion ranges of result sets, i.e. the closeness to the mean. For this measure dispersion ranges are estimated starting with the mean scores. From the mean scores one standard deviation is subtracted to define the lower limit, and one standard deviation is added to the mean score to define the upper limit. The formulae for the calculations are:

lower limit: mean – standard deviation
upper limit: mean + standard deviation

If the dispersion ranges do not overlap, then there is a significant difference between the two sets of the results. If the dispersion ranges do overlap, then there is no statistically significant difference.

According to the data in Table 13 there is no statistically significant difference between male and female results, indicating that men and women evaluated the stimuli similarly. Therefore there is no significant difference between result sets for:

1. the male listeners evaluating the original voice and the female listeners evaluating the original voice,
2. the male listeners evaluating the pseudo-female voice and the female listeners evaluating the pseudo-female voice,
3. the male listeners evaluating the pseudo-male voice and the female listeners evaluating the pseudo-male voice.

However, there is a significant difference between result sets of:

1. original and pseudo-female voice,
2. original and pseudo-male voice.

There is no significant difference between the results of pseudo-female and pseudo-male voice.

Summarising, the informal comparison shows that there was no statistically

significant difference found between male and female subjects. No statistically significant difference was found between synthetic pseudo-female and pseudo-male voices. There was found statistically significant difference between natural and artificial voices.

The second method which was adopted to compare the test results was the T-test. In this test two result sets are compared to measure the possibility of the sets being similar or different.

Type 3 of the OpenOffice T-test, meaning that there are two samples of unequal variance, was performed on the following data:

1. results of the original voice vs. results of the pseudo-female voice,
2. results of the original voice vs. results of the pseudo-male voice,
3. results of the pseudo-female voice vs. results of the pseudo-male voice.

Additionally, the results of sentences synthesised with the pseudo-female voice were compared with the results of the same sentences uttered by a human speaker. Similarly, the results of sentences synthesised with the pseudo-female voice were compared with the results of the same sentences synthesised with the pseudo-male voice. For those tests the type 1 of the OpenOffice T-test was adopted, meaning a paired test. The results are shown in Table 14.

Table 14: Results of T-test performed on the result sets of Test 2 (calculated with OpenOffice two-tailed type 3 test – unequal variance, and type 1 test – paired test).

	Pseudo-female vs. Original	*Pseudo-female vs. Pseudo-male*	*Pseudo-male vs. Original*
Type 3	0,000000	0,000223	0,000000
Type 1	0,000000	0,000011	–

The results in Table 14 show that there is probability indistinguishable from 0% of the pseudo-female voice being similar to the original voice. The same is true for the pseudo-male and the original voice. The result sets of the pseudo-female and the pseudo-male voices show that there is 99,98% chance that the results are different. Therefore, the hypothesis of the grades being equal on the significance level alpha <0,0003 is refuted.

When it comes to the paired tests, type 1 test, the probability that the result sets for the pseudo-female and the original voices are different is indistinguishable from

100%. There is 99,99% probability that the results of the pseudo-female voice will be different from the results of the pseudo-male voice.

Summarising, the results show that all the three voices were evaluated differently. Furthermore, the same sentences synthesised with the pseudo-female and pseudo-male voices did not get the same scores from the subjects. Also the synthetic stimuli produced with the pseudo-female and their original counterparts were evaluated differently. It suggests that the listeners took into account the voice characteristics, i.e. the tone of voice, while assessing the speech.

8.2.2.3.3 Results: Test 3

The test results for Test 3 are presented in Table 15. Detailed test results are shown in Appendix F.

Table 15: Test results for Test 3. Judgements given by the subjects are in rows, actual input is in columns. There were 5 items for each input category. N is the number of subjects.

		INPUT							
		Statement		*Question*		*Exclamation*		*Continuation phrase*	
JUDGEMENTS	**Polish male** *(N=8)*								
	Statement	4,00	80%	0,25	5%	4,50	90%	3,38	68%
	Question	0,00	0%	4,25	85%	0,38	8%	0,50	10%
	Don't know	1,00	20%	0,50	10%	0,13	3%	1,13	23%
	Polish female *(N=10)*								
	Statement	4,40	88%	0,30	6%	3,50	70%	3,40	68%
	Question	0,30	6%	4,30	86%	0,70	17%	0,80	16%
	Don't know	0,30	6%	0,40	8%	0,80	16%	0,80	16%
	Polish overall *(N=18)*								
	Statement	4,22	84%	0,28	6%	3,94	79%	3,39	68%
	Question	0,17	3%	4,28	86%	0,56	11%	0,67	13%
	Don't know	0,61	12%	0,44	9%	0,50	10%	0,94	19%
	Foreigners *(N=2)*								
	Statement	2,00	40%	2,50	50%	1,00	20%	1,50	30%
	Question	1,00	20%	2,50	50%	3,50	70%	1,50	30%
	Don't know	2,00	20%	0,00	0%	0,50	10%	2,00	40%

Table 15 shows the test results for Polish male and female subjects, their overall and results for the foreign subjects. In the Table the results are presented as an

average number of recognised units in one test and its percentage. The results for these groups are as follows:

1. A word at the end of a statement recognised as:
    1. A word at the end of a statement:
        1. Male subjects: 4,00 statements out of 5 recognised as statements,
        2. Female subjects: 4,40 statements out of 5 recognised as statements,
        3. Overall: 4,22 statements out of 5 recognised as statements,
        4. Foreigners: 2,00 statements out of 5 recognised as statements.
    2. A word at the end of a question:
        1. Male subjects: 0,00 statements out of 5 recognised as questions,
        2. Female subjects: 0,30 statements out of 5 recognised as questions,
        3. Overall: 0,17 statements out of 5 recognised as questions,
        4. Foreigners: 1,00 statement out of 5 recognised as questions.
    3. “Don't know”
        1. Male subjects: 1,00 statement out of 5 were not recognised,
        2. Female subjects: 0,30 statements out of 5 were not recognised,
        3. Overall: 0,61 statements out of 5 were not recognised ,
        4. Foreigners: 2,00 statements out of 5 were not recognised.
2. A word at the end of a question recognised as:
    1. A word at the end of a statement:
        1. Male subjects: 0,25 questions out of 5 recognised as statements,
        2. Female subjects: 0,30 questions out of 5 recognised as statements,
        3. Overall: 0,28 questions out of 5 recognised as statements,
        4. Foreigners: 2,50 questions out of 5 recognised as statements.
    2. A word at the end of a question:
        1. Male subjects: 4,25 questions out of 5 recognised as questions,
        2. Female subjects: 4,30 questions out of 5 recognised as questions,
        3. Overall: 4,28 questions out of 5 recognised as questions,
        4. Foreigners: 2,50 questions out of 5 recognised as questions.
    3. “Don't know”
        1. Male subjects: 0,50 questions out of 5 were not recognised,
        2. Female subjects: 0,40 questions out of 5 were not recognised,
        3. Overall: 0,44 questions out of 5 were not recognised ,

4. Foreigners: 0,00 questions out of 5 were not recognised.

3. A word at the end of an exclamation recognised as:
    1. A word at the end of a statement:
        1. Male subjects: 4,50 exclamations out of 5 recognised as statements,
        2. Female subjects: 3,50 exclamations out of 5 recognised as statements,
        3. Overall: 3,94 exclamations out of 5 recognised as statements,
        4. Foreigners: 1,00 exclamation out of 5 recognised as statements.
    2. A word at the end of a question:
        1. Male subjects: 0,38 exclamations out of 5 recognised as questions,
        2. Female subjects: 0,70 exclamations out of 5 recognised as questions,
        3. Overall: 0,56 exclamations out of 5 recognised as questions,
        4. Foreigners: 3,50 exclamations out of 5 recognised as questions.
    3. “Don't know”
        1. Male subjects: 0,13 exclamations out of 5 were not recognised,
        2. Female subjects: 0,80 exclamations out of 5 were not recognised,
        3. Overall: 0,50 exclamations out of 5 were not recognised ,
        4. Foreigners: 0,50 exclamations out of 5 were not recognised.
4. A word at the end of a continuation phrase recognised as:
    1. A word at the end of a statement:
        1. Male subjects: 3,38 continuation phrases recognised as statements,
        2. Female subjects: 3,40 continuation phrases out of 5 recognised as statements,
        3. Overall: 3,39 continuation phrases out of 5 recognised as statements,
        4. Foreigners: 1,50 continuation phrases out of 5 recognised as statements.
    2. A word at the end of a question:
        1. Male subjects: 0,50 continuation phrases recognised as questions,
        2. Female subjects: 0,80 continuation phrases out of 5 recognised as questions,
        3. Overall: 0,67 continuation phrases out of 5 recognised as questions,
        4. Foreigners: 1,50 continuation phrases out of 5 recognised as questions.
    3. “Don't know”

1. Male subjects: 1,13 continuation phrases out of 5 were not recognised,
2. Female subjects: 0,80 continuation phrases out of 5 were not recognised,
3. Overall: 0,94 continuation phrases out of 5 were not recognised ,
4. Foreigners: 2,00 continuation phrases out of 5 were not recognised.

The results for the Polish listeners show that:

1. 84% of the statement-words were recognised correctly as words at the end of a statement,
2. 86% of the question-words were recognised correctly as words at the end of a question,
3. 79% of the exclamation-words were recognised as words at the end of a statement, indicating that the intonation of these words was similar to the intonation of a statement. This result proved what was expected.
4. 68% of the continuation-phrase-words were recognised as words at the end of a statement. This result does not meet the expectations, because it was assumed that the intonational pattern of words at the end of continuation phrases will sound more like a question, than like a statement.

To sum up, the overall results for Poles of the correctly recognised statement-words and question-words indicate that the intonation in the ACCS synthesis system is very good. The exclamation-words and continuation-phrase-words were added to the test as distractors and were not the main objective of this study.

8.3 Summary – Evaluation of the ACCS synthesis system

In this chapter the evaluation methods have been discussed and results of a set of evaluation tests which were run on the ACCS synthesis system have been presented. The tests showed that:

1. Diagnostic tests:
    1. The program works correctly, but the *phoplayer*, an individual component of the program, fails to play all the automatically created PHO files when it has to handle with over 20 PHO files in a directory.

2. The program does not run correctly when it comes across an erroneous BLF annotation file, e.g. with final pause set incorrectly or wrong phoneme labels.

2. Evaluation tests:

    1. The synthetic speech produced in the ACCS procedure is highly intelligible at a word level.
    2. The quality of synthetic speech is not judged to be as high as the human recordings.
    3. The intonation of the synthetic speech is very good when it comes to the intonation of the questions and statements. The other intonational patterns were not investigated.
    4. The informal questionnaire showed that the synthetic speech is taken for a human speech with inaccurate articulation.
    5. The good results on the speech output assessment tests indicate that the annotations on the phoneme level are correct.

To sum up, the good results of the evaluation tests of the ACCS synthesis system demonstrate that ACCS synthesis may be successfully used as a method for creating speech stimuli for perception tests and as a valuable tool for checking annotations.

CHAPTER 9 Conclusion and future strategies

In this thesis, the development of a speech synthesis component for potential use in speech perception tests for children with a cochlear implant was described and a prototype implementation was developed. Use cases which served to define the deployment of the TTS software were outlined, and requirements derived from these use cases which, together with an overview of available resources, were employed in specifying the system design and in outlining future developments. Furthermore, an overview of TTS systems was provided and the reasons of choosing MBROLA as the speech synthesis engine for developing the CCS were underlined. The development procedure described here covered in detail the first two of the three planned development stages, and scarcely referring to the third Parametric CCS synthesis stage:

1. MCCS synthesis: manual format conversion from empirical data (speech recordings and time-aligned annotations) into the synthesis engine (MBROLA) interface format.
2. ACCS synthesis: automatic format conversion which emulates the manual format conversion procedure, using additional interface formats.
3. PCCS synthesis (manipulation of one parameter was performed in this study): adaptation of the pitch values extracted from the recordings of male voice to female voice.

The MCCS procedure was developed as a best case gold standard for speech synthesis with MBROLA, against which future developments would be measured. The ACCS procedure was evaluated against this benchmark, with results which were, while not numerically identical with the MCCS procedure (due to differences

in the pitch extraction procedure), indistinguishable from it in informal perception tests.

For this study, diagnostic evaluation of the ACCS method was carried out with positive results. The program's performance went very well, with one exception. All the components creating NLP-DSP interfaces automatically worked correctly, however the *phoplayer* failed to play all the files in a directory when having to handle with over 20 PHO files at runtime. This additional component needs to be looked at, but is not a part of the process creating NLP-DSP interfaces and does not interfere with ACCS procedure. Therefore, the overall results of the diagnostic evaluation are very satisfactory.

In addition, speech output assessment was performed. Different methods testing naturalness and intelligibility of synthetic speech were outlined. Speech quality tests were carried out on a representative group of male and female Polish subjects. Also two foreigners with good knowledge of Polish took part in the test. The results of the tests were presented and discussion on the tests was provided, concluding in the ACCS synthesis being highly intelligible, but not accepted as well as the speech produced by a professional speaker.

Finally, the evaluation tests served to validate the annotated speech corpus on which the CCS system is based. The results of the diagnostic tests showed some errors in the BLF files, which had to be corrected automatically or manually. (The erroneous annotation files were not used for creating stimuli for the speech output assessment tests.) The speech perception tests, on the other hand, showed that the synthetic speech is highly intelligible, which indicated that the annotations were done correctly.

In the future the third stage, a Parametric Close Copy Speech (PCCS) synthesis procedure, will be developed, with the ACCS procedure as the platform for parametrising the prosodic features. Work on the PCCS procedure is in progress.

At a later stage, the application of these procedures to unit selection speech synthesis is planned.

Software

ActivePerl. 1996-2006 ActiveState Software Inc.

Boersma, P. & Weenink, D. 2001. PRAAT, a system for doing phonetics by computer. Glot International 5(9/10): 341-345.

Dutoit, T. 2005. The MBROLA Project. January 4th, 2005. <http://www.tcts.fpms.ac.be/synthesis/mbrola.html>, accessed 2006-10-15.

ATIP GbR, Advanced Technologies for Information Processing. 1998. DE2 - A German Male Voice. Copying the MBROLA Bin and Databases. January 3rd, 2005. <http://tcts.fpms.ac.be/synthesis/mbrola/mbrcopybin.html>, accessed 2006-10-15.

Lennes, M. 2003. Praat script – collect_pitch_data_from_files.praat. <http://www.helsinki.fi/~lennes/praat-scripts/public/collect_pitch_data_from_files.praat>, accessed 2006-02-18.

Portele, T. 1999. IKR Forschung: Phonetik Txt2Pho. July 4th, 2000. <http://www.ikp.uni-bonn.de/dt/forsch/phonetik/hadifix/HADIFIXforMBROLA.html>, accessed 2006-10-15.

Szklanny, K. & Masarek, K. 2002. PL1 - A Polish female voice for the MBROLA synthesizer. Copying the MBROLA Bin and Databases. <http://tcts.fpms.ac.be/synthesis/mbrola/mbrcopybin.html>, accessed 2006-11-25.

Sjölader, K. & Jonas, B. 2005. WaveSurfer 1.8.5/0511011429 © 2005.

Bibliography

Ashby, A. & Maidment, J. 2005 *Introducing Phonetic Science.* Cambridge: Cambridge University Press.

Bachan, J. 2006. Verification of a Set of Speech Perception Tests for Children with a Cochlear Implant. Speech signal annotation, processing and synthesis. In: *Proceedings of Speech Signal Annotation, Processing and Synthesis Symposium*, Poznań, Poland.

Bachan, J. & Gibbon, D. 2006. Close Copy Speech Synthesis for Speech Perception Testing. In: *Investigationes Linguisticae*, vol. 13, pp. 9-24. <http://www.staff.amu.edu.pl/~inveling/pdf/Jolanta_Bachan_Dafydd_Gibbon_INVE13.pdf>

Black, A. W. 2002. Perfect Synthesis for all of the people all of the time. In: *IEEE TTS Workshop 2002.* Santa Monica, CA.

Black, A. W. & Lenzo, K. 2000. Limited domain synthesis. In: *ICSLP2000*, Beijing, China.

Black, A. W. & Taylor, P. 1997. Automatically clustering similar units for unit selection in speech synthesis. In: *Proceedings of the Eurospeech 1997*, vol. 2, pp. 601-604. Rhodes, Greece.

Boersma, P. & Weenink, D. 2001. PRAAT, a system for doing phonetics by computer. Glot International 5(9/10), pp. 341-345.

BOSS, the Bonn Open Synthesis System. <http://www.ikp.uni-bonn.de/boss/>, accessed 2007-03-15.

Campbell, N. & Black, A. W. 1995. Prosody and the selection of source units for concatenative synthesis. In: van Saten, J. & Sproat, R. & Olive, J. & Hirschberg, J., editors, *Progress in Speech Synthesis*. Springer Verlag.

Demenko, G., Wypych, M. & Baranowska, E. 2003. Implementation of Grapheme-to-Phoneme Rules and Extended SAMPA Alphabet in Polish Text-to-Speech Synthesis. *Speech and Language Technology*, vol. 7. Poznań: Zakład Graficzny UAM.

Demenko, G., Grocholewski, S., Wagner, A., Szymanski M. 2006. Prosody annotation for corpus based speech synthesis. In: *Proceedings of the Eleventh Australasian International Conference on Speech Science and Technology*, pp. 460-465. Auckland, New Zealand.

Donovan, R. E. 1996 *Trainable Speech Synthesis*. Ph.D. Dissertation. Cambridge University Engineering Department, England.

Dutoit, T. 1997. *An Introduction To Text-To-Speech Synthesis*. Dordrecht: Kluwer Academic Publishers.

Dutoit, T. 2005. The MBROLA project. <http://www.tcts.fpms.ac.be/synthesis/mbrola.html>, accessed 2006-11-30.

Dutoit, T., Pagel, V., Pierret, N., Bataille, F. & van der Vrecken O. 1996. The MBROLA Project: Towards a Set of High-Quality Speech Synthesizers Free of Use for Non-Commercial Purposes. In: *Proceedings of ICSLP 96*. Philadelphia, vol. 3, pp. 1393-1396.

Engwall, O. 1998. A 3D vocal tract model for articulatory and visual speech synthesis. <http://www.speech.kth.se/~olov/papers/fonetik98.pdf>, accessed 2007-03-11.

Festival, the Festival Speech Synthesis System. The Centre for Speech Technology Research, the University of Edinburgh. <http://www.cstr.ed.ac.uk/projects/festival/>, accessed 2007-03-15.

Gibbon, D., Moore, R. & Winski, R. 1997. *Handbook of Standards and Resources for Spoken Language Systems*. Berlin: Mouton de Gruyter.

Gibbon, D., Mertins, I. & Moore, R. 2000. *Handbook of Multimodal and Spoken Dialogue Systems: Terminology, Resources and Product Evaluation*. New York: Kluwer Academic Publishers.

Goldstein, M. 1995. Classification of methods used for assessment of text-to-speech systems according to the demands placed on the listener. In: *Speech Communication*, vol. 16, pp. 225-244.

Gut, U. & Milde, J-T. 2003. Annotation and Analysis of Conversational Gestures in the TASX environment. *Künstliche Intelligenz* 17:4.

Hammerl, R. & Sambor, J 1990, *Statystyka dla językoznawców* [Statistics for linguists]. Warszawa: PWN.

Hawkins, S., Heid, S., House, J. & Huckvale, M. 2000. Assessment of naturalness in the Prosynth Speech Synthesis Project. In: *IIEE: State-of-the-art in speech synthesis.* 13/1-13/6. London, Great Britain.

Hunt, A. & Black, A. W. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In: *Proceedings of ICASSP 96*, vol 1, pp. 373-376. Atlanta, Georgia.

Jassem, W. 2003. Polish. In: *Journal of the International Phonetic Association*, vol. 33, pp: 103-107. Cambridge: Cambridge University Press.

Jekosch, U. 2005. *Voice and Speech Quality Perception: Assessment and Evaluation.* Berlin: Springer.

Jekosch, U., & Pols, L. 1994. A feature-profile for application-specific speech synthesis assessment and devaluation, In: *Proceedings of the 3rd International Conference on Spoken Language Processing*, Yokohama.

Jurafsky, D. & Martin, J. H. 2000. *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Klabbers, E., Stöber, K., Veldhuis, R, Wagner, P. & Breuer, S. 2001. Speech Synthesis Development Made Easy: the Bonn Open Synthesis System. In: *Eurospeech-2001*, pp. 521-525.

Klatt, D. H. 1987. Review of Text-To-Speech Conversion for English. In: *Journal of the Acoustical Society of America* 82 (3), pp. 737-793.

Lammetty, S. 1999. *Review of Speech Synthesis Technology*. M.A. thesis. Department of Electrical and Communication Engineering. Helsinki University of Technology.

Lampert, A. 2004. Evaluation of the MU-TALK Speech Synthesis System. <http://www.ict.csiro.au/staff/Andrew.Lampert/writing/SynthesisEvaluation.pdf >, accessed 2007-02-12.

Łobacz, P. 2002. Badania fonostatyczne na potrzeby syntezy mowy. Phonostatistical Research for Speech Synthesis. In: *Speech and Language Technology, vol. 6.* pp. 81-112.

Mattingly, I. G. 1974. Speech Synthesis for Phonetic and Phonological Models. In: *Current Trends in Linguistics*, vol. 12, pp. 2451-2487. The Hague: Mouton.

Ng, K. 1998. *Survey of Data-Driven Approaches to Speech Synthesis.* Spoken Language Systems Group, Massachusetts Institute os Technology. <http://www.speech.tuc.gr/docs/ext/tts/intro/kng-1998-data_driven_survey.pdf>, accessed 2007-03-12.

Ogórkiewicz, J., Bachan, J., Mazur, M., Komar, M. & Demenko, G. 2005. A set of speech perception tests for children with cochlear implants – preliminary evaluation. In *Proceedings of Speech, Analysis, Synthesis and Recognition Symposium*, Kraków, Poland.

Pallett, D. S. & Fourcin A. 1996. Speech Input: Assessment and Evaluation. In: Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A. & Zue, V., editors, *Survey of the State of the Art in Human Language Technology.* Cambridge: Cambridge University Press, and Pisa: Giardini Editori e Stampatori.

Ryalls, J. 1996. *A Basic Introduction to Speech Perception*. San Diego, California: Singular Publishing Group, Inc., and London: Singular Publishing Ltd.

Schwartz, R. L., Olson, E. & Christiansen, T. 1997. *Learning Perl on Win32 Systems.* Sebastpol: O'Reilly & Associates, Inc.

Steffen-Batogowa, M. 1975. *Automatyzacja transkrypcji fonematycznej tekstów polskich* [Automatic phonemic transcription of Polish texts]. Warszawa: Państwowe Wydawnictwo Naukowe.

Stevens, K. N. & Bickley, C. A. 1990. Higher-Level Control Parameters for a Formant Synthesiser. In: *SSW1-1990*, pp. 63-66.

Styger, T., & Keller, E. 1994. Formant synthesis. In E. Keller (ed.), *Fundamentals of Speech Synthesis and Speech Recognition: Basic Concepts, State of the Art, and Future Challenges,* pp. 109-128. Chichester: John Wiley.

Szklanny, K. & Masarek, K. 2002. PL1 - A Polish female voice for the MBROLA synthesizer. Copying the MBROLA Bin and Databases. <http://tcts.fpms.ac.be/synthesis/mbrola/mbrcopybin.html>, accessed 2006-11-25.

Sjölader, Kåre & Jonas Beskow. 2005. WaveSurfer 1.8.5/0511011429.

Van Bezooijen, R. & Pols, L. 1990. Evaluating text-to-speech systems: Some methodological aspects. In: *Speech Communication,* vol. 9, pp. 263-270.

Taylor, P. A., Black, A. & Caley, R. 1998. The architecture of the Festival speech synthesis system. In: *The Third ESCA Workshop in Speech Synthesis*, pp. 147-151, Jenolan Caves, Australia.

Wall, L., Christiansen, T. & Schwartz, R. L. 1996. *Programming Perl.* Sebastpol: O'Reilly & Associates, Inc.

Wikipedia, the free encyclopedia. <http://www.wikipedia.org/>

Appendix A Speech perception tests for children with a cochlear implant

NONSENSE STIMULI TEST

PRELIMINARY LEVEL

Discrimination of quantity

Discrimination of isolated phones

Discrimination of rhythm

Discrimination of intonation

Identification of amplitude

Identification of voice

Identification of vowels

Identification of consonants

Identification of suprasegmental characteristics - intonation

VERBAL STIMULI TEST[10]

C. CLOSED VERBAL TESTS – NATURAL SPEECH

C0. 0 LEVEL – BASIC

C01. SEGMENTAL PERCEPTION

C00. Introduction of the lexicon*[11]

C01. Identification of disyllabic words – younger children

C02. Identification of disyllabic words – older children

C02. SUPRASEGMENTAL PERCEPTION

C03. Identification of unstressed syllables – younger children

10 The numbering type of the tests is taken from the set of tests.

11 The tests marked with an asterisk did not work when the tests were examined. Work on them is in progress.

C04. Identification of unstressed syllables – older children

C05. Identification of voice

C1. 1 LEVEL – MEDIUM

C11. SEGMENTAL AND SUPRASEGMENTAL PERCEPTION

C11. Discrimination of rhythmic patterns and identification of mono- and disyllable words – easy version

C12. Discrimination of rhythmic patterns and identification of mono- and disyllable words – difficult version

C13. Identification of thrisyllabic words

C14. Discrimination of rhythmic patterns and identification of mono-, di-, tri- and four-syllable words

C12. SUPRASEGMENTAL PERCEPTION

C15. Identification of suprasegmental characteristics in two-word phrases

C16. Identification of suprasegmental characteristics in three-word phrases

C13. SEGMENTAL PERCEPTION OF CHARACTERISTICS

C17. Identification of segmental characteristics – vowels – easy version

C18. Identification of segmental characteristics – vowels – difficult version

C19. Identification of segmental characteristics – consonants – easy version

C20. Identification of segmental characteristics – consonants – difficult version

C21. Identification of segmental characteristics in words of minimal contrast supported by visual information

C22. Identification of segmental characteristics in words and logatoms

C23. Perception of basic phonetic-acoustic structures of the Polish language – easy version

C24. Perception of basic phonetic-acoustic structures of the Polish language – difficult version

C25. Identification of segmental characteristics*

C2. 2 LEVEL – ADVANCED

C21. TESTS OF AUDITORY MEMORY

C26. Memorisation of units

C27. Memorisation of linguistic structure

C22. SUPRASEGMENTAL RECOGNITION

C28. Intonation – questions, statements*

C29. Intonation – questions, statements – adjectives*

C23. SEGMENTAL RECOGNITION

C30. Recognition of phrases – thematic tests – „Dzień Malucha (“A child's day”)

C31. Recognition of phrases – thematic tests – „Poszukaj Zwierzątka” (“Find the animal”)

C32. Recognition of phrases – thematic tests – „Pan Ziemniak (“Mr Potato”)

D. SPEECH RECOGNITION IN THE OPEN SET

D0. RECOGNITION OF WORDS

D01. RECORNITION OF WORDS

D01. Recognition of words – disyllabic words*

D02. Recognition of words – monosyllabic words*

D02. RECOGNITION OF WORDS IN PHRASES

D03. „Dzień Malucha 2” (“A child's day 2”)

D04. „Pan Ziemniak 2” (“Mr Potato 2”)*

D1. TESTS OF AUDITORY MEMORY

D05. Memorisation of units

D06. Memorisation of units, complex linguistic structure

E. RECOGNITION AND INTELLIGIBILITY OF SPEECH IN THE OPEN SET

E0. RECOGNITION AND INTELLIGIBILITY OF SPEECH

E01. RECOGNITION OF SIMPLE PHRASES

E01. Recognition and intelligibility of simple phrases – easy version*

E02. Recognition and intelligibility of simple phrases – difficult version*

E02. RECOGNITION OF COMPLEX PHRASES

E03. Recognition and intelligibility of complex phrases based on key words

E04. Recognition and intelligibility of complex phrases – easy version

E05. Recognition and intelligibility of complex phrases – difficult version

E1. RECOGNITION OF CONTINUOUS SPEECH

E06. „Kasia pomaga mamie" ("Kasia helps her mother")

E07. „Krzyś jest głodny" ("Krzyś is hungry")

E08. „Jaś jest chory" ("Jaś is ill")

E09. „Dziadek Gosi mieszka na wsi" ("Gosia's grandpa lives in a village")

Appendix B *BLF2PHO* Perl script

```
# 02-01-2007
# Jolanta Bachan

$directory = "." ;

$extension01 = ".blf" ;
$extension02 = ".pho" ;
$extension03 = ".TextGrid" ;
$extension04 = ".pitch";
$add = "_pitch";
$i=0 ;

system("del *$extension02" > a);
system("del *$extension03" > a);
system("del *$extension04" > a);

# ------------------------------------------------------------------
# DIR loop
# State the directory and list the files that are stored there

opendir (NT, "$directory") || die "Cannot opendir $directory: $!";


# Write the names of the files that are in the directory as a list
of filenames in a TXT document

open (OUTPUT, ">filenames.txt") || die "Cannot create
filenames.txt:$!";
foreach $name(sort readdir(NT)) {
      print OUTPUT "$name\n";
}
close(OUTPUT);
closedir(NT);

#------------------------------------------------------------------
# Put the filenames in an array

open (INPUT, "filenames.txt");
```

```
while (<INPUT>) {
      @old = (@old, $_);
}
close(INPUT);

@new = @old;

# -----------
# $a stores the number of the elements that are in the @new and
# @old, since the arrays are identical so far

$a = @new;

#------------
# Change the elements of the @new array

for ($i=0; $i<$a; $i++) {
      $new[$i] =~ s/$extension01/$extension02/g;
}

# ------------------------
# Check the element of the @old array and if their extension
# matches BLF, then
# open it, create a twin file.

for ($i=0; $i<$a; $i++) {

      if ($old[$i] =~ /$extension01/) {
            open (IN, $old[$i]) || die "Cannot open $old[$i]:$!";
            open (OUT, ">$new[$i]") || die "Cannot create
$new[$i]:$!";

# Read the input (old) file and copy the content, making the
# substitution, into the twin file.

#-----------------------------------------------------------------
#-----------------------------------------------------------------
# blf2pho_polish-monotone

$samplerate = 16 ;
```

```
$pitchpos = 50 ;
$pitch = 200 ;

$previous = 0 ;
$previous02 = 0 ;
$previous03 = 0 ;

$initphoneme = "_" ;
$initduration = 200 ;

$previousphoneme = $initphoneme ;

            $eovowel = "" ;

#-------------------------------------------------------------------
# The Loop converting BLF to PHO format

while (<IN>) {

$sample_3 = $sample_2 ;
$sample_2 = $sample_1 ;
$sample_1 = $sample ;

($sample, $phoneme, $prosody) = split;

$_ = $previousphoneme ;
$previousphoneme = $phoneme ;

#--------------------------------------
# Convert sample rate to milliseconds

$millisec = int($sample / $samplerate) ;
$millisec02 = int($sample_1 / $samplerate) ;
$millisec03 = int($sample_2 / $samplerate) ;

#--------------------------------------
# Convert millisecond pairs to durations

$duration = $millisec - $previous ;
$duration02 = $millisec02 - $previous02 ;
```

```
$duration03 = $millisec03 - $previous03 ;

#-------------------------------------
# Save millisecond value for next time

$previous = $millisec ;
$previous02 = $millisec02 ;
$previous03 = $millisec03 ;

#--------------------------------

s/\$j(.+)/\1/g ;
s/_(.+)/\1/g ;
s/\$[pj]/_/g ;
s/\?/_/g ;
s/[#`.^"%*&\/|:]//g ;
s/y/I/g;
s/c/k/g;
s/J/g/g;

if ($eovowel ne "") {
      if (/w~/ || /j~/) {
            s/[wj]/$eovowel/;
            $duration_sum=$duration02+$duration;
            $nasal_vowel=$_;
            print OUT $nasal_vowel . "\t" . $duration_sum . "\t" .
$pitchpos . "\t" . $pitch . "\n" ;
            $nasal_vowel = "" ;
            $_ = "" ;
      } elsif (/[oe]/) {
            $simplevowel_first = $eovowel ;

            print OUT $simplevowel_first . "\t" . $duration02 . "\t"
. $pitchpos . "\t" . $pitch . "\n" ;

            $simplevowel_first = "" ;

      } else {
            $simplevowel = $eovowel ;
            }
```

```
            $eovowel = "" ;
      }


if ($_ eq "e" || $_ eq "o") {
      $eovowel = $_ ;
      $simplesample = $sample ;
      $_ = "" ;
      }


s/\@/e/g;

#--------------------------------------
# Printing conditions

if ($_ ne "") {
      if ($simplevowel ne "" ) {
            print OUT $simplevowel . "\t" . $duration02 . "\t" .
$pitchpos . "\t" . $pitch . "\n" ;
            $simplevowel = "" ;
            $simplesample = "" ;
      }

      print OUT $_ . "\t" . $duration . "\t" . $pitchpos . "\t" .
$pitch . "\n" ;

      }
}

if ($eovowel ne "") {
      print OUT $eovowel . "\t" . $duration . "\t" . $pitchpos .
"\t" . $pitch ."\n";
      $eovowel = "" ;
}
print OUT $initphoneme . "\t" . $initduration . "\t" . $pitchpos .
"\t" . $pitch . "\n" ;

#---------------------------------------------------------------
#---------------------------------------------------------------
```

```
            close (IN);
            close (OUT);
      }
}

#---------------------------------------------------------------------
# Create a TextGrid file

$file_type = "\"ooTextFile\"" ;
$object_class = "\"TextGrid\"" ;
$min = 0 ;
$max = 0 ;   # duration of the whole file
$tier = "tiers? <exists>";
$item = 1;   # the number of tiers
$n = 1 ;
$class = "\"IntervalTier\"";
$name = "\"phonemes\"" ;
$size = 0 ;  # the number of intervals
$intervals = 0 ;  # the number of the interval
# $xmin = "" ;
# $xmax = "" ;
$text = "";

$max_msec = 0 ;
$xmin_sec = 0 ;
$xmax_sec = 0 ;
@xmax = "" ;
@xmin = "" ;

@TGrid = @new ;

for ($i=0; $i<$a; $i++) {

      $TGrid[$i] =~ s/$extension02\b/$extension03/g;
}

#print @TGrid;

for ($i=0; $i<$a; $i++) {
```

```
      if ($new[$i] =~ /$extension02\b/) {

            open (INP, $new[$i]) || die "Cannot open $new[$i]:$!";
            open (OUTP, ">$TGrid[$i]") || die "Cannot create
$TGrid[$i]:$!";

#-------------------------------------------------------------------
# The Loop converting PHO to TextGrid format

 @text = "" ;
# $size = 0 ;
# $xmin_msec = 0 ;
# @xmin = "" ;
 $max_msec;
 $seconds = 0 ;
# @xmax = "" ;
 $length = 0 ;
# $size = 0 ;
 $max = 0 ;
# $j = 0 ;

$file_type = "\"ooTextFile\"" ;
$object_class = "\"TextGrid\"" ;
$min = 0 ;
$max = 0 ;   # duration of the whole file
$tier = "tiers? <exists>";
$item = 1;   # the number of tiers
$n = 1 ;
$class = "\"IntervalTier\"";
$name = "\"phonemes\"" ;
$size = 0 ;  # the number of intervals
$intervals = 0 ;  # the number of the interval
@text = "";

$max_msec = 0 ;
$xmin_sec = 0 ;
$xmax_sec = 0 ;
@xmax = "" ;
@xmin = "" ;
```

```
while (<INP>) {
      ($phonemes, $durations, $pitch_position, $pitch_value) =
split;

      @text = (@text, $phonemes) ;
      $xmin_msec = $seconds;
      @xmin = (@xmin, $xmin_msec) ;

# ------------
# max_msec stores the duration of the whole utterance in
# milliseconds

      $max_msec = $max_msec + $durations ;
      $seconds = ($max_msec)/1000;
      @xmax = (@xmax, $seconds);
}

$length = @text ; # the number of phonemes in the file
$size = ($length-3) ; # get rid of the additional phonemes
                      # generated and one additional line from
                      # 'nowhere' automatically

$max = ($max_msec-200)/1000 ; # get rid of the automatically added
                              # 200msec for a pause at the end of
                              # a PHO file

#---------------------------------------------------------------------
# Print the TextGrid file

print OUTP "File type = $file_type" . "\n" ;
print OUTP "Object class = $object_class" . "\n" . "\n";


print OUTP "xmin = $min\n" ;
print OUTP "xmax = $max\n" ;
print OUTP $tier . "\n" ;
print OUTP "size = $item\n" ;
print OUTP "item []:\n" ;

print OUTP "\t" . "item [1]:\n" ;
```

```
print OUTP "\t" . "\t" . "class = $class\n" ;
print OUTP "\t" . "\t" . "name = $name\n" ;
print OUTP "\t" . "\t" . "xmin = $min\n" ;
print OUTP "\t" . "\t" . "xmax = $max\n" ;
print OUTP "\t" . "\t" . "intervals: size = $size\n" ;

for ($j=2; $j<=$size; $j++) {
      $m = $j-1;
print OUTP "\t" . "\t" . "intervals [$m]:\n" ;
print OUTP "\t" . "\t" . "\t" . "xmin = $xmin[$j]\n" ;
print OUTP "\t" . "\t" . "\t" . "xmax = $xmax[$j]\n" ;
print OUTP "\t" . "\t" . "\t" . "text = \"$text[$j]\"\n" ;
}
      if ($j = $size) {
      $p = $j+1;
            print OUTP "\t" . "\t" . "intervals [$j]:\n" ;
            print OUTP "\t" . "\t" . "\t" . "xmin = $xmin[$p]\n" ;
            print OUTP "\t" . "\t" . "\t" . "xmax = $xmax[$p]\n" ;
            print OUTP "\t" . "\t" . "\t" . "text = \"$text[$p]\"\n"
;
      $p=0;
      }
$m=0;
$j=0;
      }

      close (INP);
      close (OUTP);
}
unlink ("filenamescopy.txt");

#---------------------------------------------------------------------
# Create pitch files

#-------
# Call the praat script

$command01 = "praatcon.exe max_pitch02.praat";
system($command01) && die "Cannot execute $command01.";
```

```
@lines = "";
$length02 = 0 ;
$i = 0 ;
$filename = "" ;
$label = "" ;
$pitch_value = "" ;
$extension = ".pitch" ;
$name = "" ;

open (IN, "pitchresults.txt") || die "Cannot open
pitchresults.txt:$!";
while (<IN>) {
      @lines = (@lines, $_) ;
      ($filename, $label, $pitch_value) = split ;
      @filenames = (@filenames, $filename) ;
      @labels = (@labels, $label) ;
      @pitch_values = (@pitch_values, $pitch_value) ;
}
close(IN);

$length02 = @lines;

for ($i=1; $i<=$length02; $i++) {
      if ($filenames[$i-1] = $filenames[$i]) {
            $file = $filenames[$i] . $extension;
            open (OUT, ">>$file") || die "Cannot create $file:$!";
            if ($file =~ /$filenames[$i]/ )     {
                  print OUT $filenames[$i] . "\t" . $labels[$i] .
"\t" . $pitch_values[$i] . "\n" ;
            }
}
close (OUT);
}

unlink ("pitchresults.txt");

#---------------------------------------------------------------------
$name = "" ;
```

```
# --------------------------
# DIR loop
# State the directory and list the files that are stored there

opendir (NT, "$directory") || die "Cannot opendir $directory: $!";

# Write the names of the files that are in the directory as a list
of filenames in a TXT document

open (OUTPUT, ">files.txt") || die "Cannot create files.txt:$!";
foreach $name(sort readdir(NT)) {
      print OUTPUT "$name\n";
}
close(OUTPUT);
closedir(NT);

#--------------------------
@pitch_files = () ;
@pho_files = () ;

open (INPUT, "files.txt") || die "Cannot open files.txt:$!";
while (<INPUT>) {
      if (/$extension02\b/) {
            @pho_files = (@pho_files, $_);
      }
      if (/$extension04\b/) {
            @pitch_files = (@pitch_files, $_);
      }
}
close(INPUT);

#-----------
# Set the variables

@new_pho_files = () ;
$addition = "_pitch";
$j=0 ;
$d = 0 ;

#-----------
```

```
# Copy the elements of @pitch_files list into a @new list

@new_pho_files = @pho_files ;

#-----------
# Get the length of the @pho_files list - the number of files in the
directory

$d = @pho_files;

#-----------------------------------------------------------------
# Main FOR loop
# - opens files listed in @pho_files and @pitch_files lists

for ($j=0; $j<$d; $j++) {

#----------
# Empty the lists

@pitch = () ;
@pitch_values02 = () ;
@mbrola = () ;
@phonemes = () ;
@durations = () ;
@pitch_positions = () ;
$a=0 ;
$x = "" ;
$y = "" ;
$z = "" ;
$s = "" ;
$k = "" ;
$l = "" ;
$m = "" ;
$n = "" ;

#---------
# Change the elements of @new list by adding "_pitch" to each
element.

$new_pho_files[$j] =~ s/$extension02/$addition$extension02/g;
```

```
#---------
# Open a pitch file

open (INPitch, "$pitch_files[$j]") || die "Cannot open
$pitch_files[$j]:$!" ;

#---------
# WHILE loop writes the content of the pitch file into a list
# and splits every line into three columns. Create a list with
# pitch values out of pitch value column.

while (<INPitch>) {
      @pitch = (@pitch, $_);
      ($filename, $label, $pitch_value02) = split;
            @pitch_values02 = (@pitch_values02, $pitch_value02) ;
}

#--------
# Close the filehandle

close(INPitch);


#-------
# Get the length of the @pitch list - the length of the pitch file.

$a = @pitch;

#---------
# Open a pho file

open (INMbrola, "$pho_files[$j]") || die "Cannot open
$pho_files[$j]:$!";


#---------
# WHILE loop writes the content of the pho file into a list and
# splits every line into four columns. Creat three lists with
# phonemes, durations and pitch positions out of these columns
```

```
while (<INMbrola>) {
      @mbrola = (@mbrola, $_);
      ($phoneme, $duration, $pitch_pos, $pitch_value01) = split;
            @phonemes = (@phonemes, $phoneme);
            @durations = (@durations, $duration);
            @pitch_positions = (@pitch_positions, $pitch_pos);
}

#--------
# Close the filehandle

close(INMbrola);

#-------
# Make the pitch file and pho file equal by removing the first and
# the last lines of the pho file. SHIFT removes the first
# element of a list. POP removes the last element of a list.

$x = shift(@phonemes);
$y = shift(@durations);
$z = shift(@pitch_positions);
$s = shift(@mbrola);

$k = pop(@phonemes);
$l = pop(@durations);
$m = pop(@pitch_positions);
$n = pop(@mbrola);

$i=0 ;

#--------
# Open an empty pho_pitch file

open (OUP, ">$new_pho_files[$j]") || die "Cannot open
$new_new_pho_files[$j]:$!";

#-------------------------------------
# In every opened pho_pitch file print the phonemes, durations
# <AND pitch positions (from the pho file) and pitch values
```

```
# (from the pitch file) if the condition is met>.

for ($i=0; $i<$a; $i++){

      if ( $pitch_values02[$i] eq "--undefined--") {
            print OUP $phonemes[$i] . "\t" . $durations[$i] ."\n";
      } else {

            print OUP $phonemes[$i] . "\t" . $durations[$i] ."\t" .
$pitch_positions[$i] . "\t" .  $pitch_values02[$i]*2 . "\n" ;
      }
}

#--------
# Close the filehandle

close(OUP);

#--------
# Delete the auxiliary txt file

unlink ("files.txt");

#-----------------------------------------------------------------
# Close the main FOR loop
}

#-----------------------------------------------------------------
# Play the pho files

$i=0 ;

for ($i=0; $i<$a; $i++) {

      if ($new_pho_files[$i] =~ /$add$extension02\b/) {
            $command = "phoplayer database=pl1 $new_pho_files[$i]";
            system($command) && die "Cannot execute $command.";
            print $new_pho_files[$i];
      }
}
```

```
# ----------
# Delete the auxiliary file and its twin brother

unlink ("filenames.txt");
system("del *$extension02");
system("del *$extension03");
system("del *$extension04");
```

Appendix C *max_pitch* Praat script

```
# This script goes through sound and TextGrid files in a  directory,
# opens each pair of Sound and TextGrid, calculates the pitch
maximum
# of each labeled interval, and saves results to a text file.
# To make some other or additional analyses, you can modify the
script
# yourself... it should be reasonably well commented! ;)
#
# This script is distributed under the GNU General Public License.
# Copyright 4.7.2003 Mietta Lennes
# Modified by Dafydd Gibbon 28-12-2006

sound_directory$ = ""
textGrid_directory$ = ""
sound_file_extension$ = ".wav"
textGrid_file_extension$ = ".TextGrid"
resultfile$ = "pitchresults.txt"
tier$ = "phonemes"
time_step = 0.01
minimum_pitch = 75
maximum_pitch = 300



# Here, you make a listing of all the sound files in a directory.
# The example gets file names ending with ".wav" from C:\tmp\

Create Strings as file list... list
'sound_directory$'*'sound_file_extension$'
numberOfFiles = Get number of strings

# Check if the result file exists:
if fileReadable (resultfile$)
      pause The result file 'resultfile$' already exists! Do you
want to overwrite it?
      filedelete 'resultfile$'
endif

# Write a row with column titles to the result file:
```

```
# (remember to edit this if you add or change the analyses!)

titleline$ = "Filename  Segment label  Maximum pitch
(Hz)'newline$'"
fileappend "'resultfile$'" 'titleline$'

# Go through all the sound files, one by one:

for ifile to numberOfFiles
      filename$ = Get string... ifile
# create a file called filename
      # A sound file is opened from the listing:
      Read from file... 'sound_directory$''filename$'
      # Starting from here, you can add everything that should be
      # repeated for every sound file that was opened:
      soundname$ = selected$ ("Sound", 1)
      To Pitch... time_step minimum_pitch maximum_pitch
      # Open a TextGrid by the same name:
      gridfile$ =
"'textGrid_directory$''soundname$''textGrid_file_extension$'"
      if fileReadable (gridfile$)

            Read from file... 'gridfile$'
            # Find the tier number that has the label given in the
form:
            call GetTier 'tier$' tier
            numberOfIntervals = Get number of intervals... tier
            # Pass through all intervals in the selected tier:
            for interval to numberOfIntervals
                  label$ = Get label of interval... tier interval
                  if label$ <> ""
                        # if the interval has an unempty label, get
its start and end:
                        start = Get starting point... tier interval
                        end = Get end point... tier interval
                        # get the Pitch maximum at that interval
                        select Pitch 'soundname$'
                        pitchmax = Get maximum... start end Hertz
Parabolic
                        # printline 'pitchmax'
```

```
                        # Save result to text file:
                        resultline$ = "'soundname$'   'label$'
      'pitchmax:0''newline$'"
                        fileappend "'resultfile$'" 'resultline$'
                        select TextGrid 'soundname$'
                  endif
            endfor
            # Remove the TextGrid object from the object list
            select TextGrid 'soundname$'
            Remove
      endif
      # Remove the temporary objects from the object list
      select Sound 'soundname$'
      plus Pitch 'soundname$'
      Remove
      select Strings list
      # and go on with the next sound file!
endfor

Remove


#-------------
# This procedure finds the number of a tier that has a given label.

procedure GetTier name$ variable$
        numberOfTiers = Get number of tiers
        itier = 1
        repeat
                tier$ = Get tier name... itier
                itier = itier + 1
        until tier$ = name$ or itier > numberOfTiers
        if tier$ <> name$
                'variable$' = 0
        else
                'variable$' = itier - 1
        endif

      if 'variable$' = 0
            exit The tier called 'name$' is missing from the file
```

```
'soundname$'!
      endif

endproc
```

Appendix D Test material

Test 1

	Filename	*Sentence*	*Words*
1	B0055	Lecz są gzymsy albo gzik.	5
2	A0150	To najsprzeczniejsze zeznanie, jakie kiedykolwiek słyszałem.	6
3	B0080	Widzą chrzan biały na rzęsach.	5
4	A0280	Wszystkie dzieci kochają wakacje.	4
5	A0210	Jego dzisiejszy występ niewątpliwie potwierdził jego ogromny talent.	8
6	A0033	Odgnieciony ślad głowy był wyraźnie na poduszce.	7
7	A0060	Wszedłszy do biura, spostrzegłam, że ktoś grzebał w moich dokumentach.	10
8	B0030	Książę daj pół ziemi i siostrę.	6
9	B0113	Najjaśniej gada z tą lalką dziś.	6
10	B0105	Tęsknił żigolak pod żlebem.	4
11	A0310	Wejście do budynku jest wzbronione.	5
12	A0250	W nocy spadła świeża warstwa śniegu i poranny krajobraz wyglądał olśniewająco.	11
13	B0060	Lecz późną nocką idą raźniej.	5
14	A0240	Dziewka umyła gliniane garnczki w strumyku i położyła na zielonej trawie do wyschnięcia.	13
15	B0075	Wal po tym czymś stopą.	5
16	B0015	Boś cały w wiśniowym soku.	5
17	A0360	Nie znam się na literaturoznawstwie.	5
18	B0045	Obcy ptak co drzemał na pniu.	6
19	A0341	Wyszłam na spacer z psem półzmierzchem.	6
20	B0005	Móc czuć każdy odczynnik.	4
		Total	*126*

Test 2

	Filename	*Sentence*	*Voice*
1	E0461	To śmieszne, że twój duży pies przestraszył się tak małej myszki.	Original
2	E0459	Wolał pozostać uczciwym robotnikiem bez kasy niż wspinać się po szczeblach kariery oszukując ludzi.	Pseudo-female
3	E0440	Herbata to ulubiony napój na śniadanie wśród wielu Polaków.	Pseudo-male
4	E0481	Dermatolog nie dał mi gwarancji, że ten nowy krem nie spowoduje wysypki na mojej skórze.	Pseudo-male
5	E0501	To małe, martwe zwierzę, które mama znalazła na chodniku było prawdopodobnie ofiarą tegorocznego mrozu.	Pseudo-male
6	E0480	Mój romantyczny brat powiedział ostatnio swojej dziewczynie, że dla niej jest w stanie postarać się nawet o gwiazdkę z nieba.	Pseudo-female
7	E0490	Marek jest bardzo wrażliwym i czułym mężczyzną, który wspaniale opiekuje się swoją rodziną.	Original
8	E0450	Dobrze by było, gdyby przeczytał wszystkie lektury z epoki pozytywizmu.	Pseudo-female
9	E0501	To małe, martwe zwierzę, które mama znalazła na chodniku było prawdopodobnie ofiarą tegorocznego mrozu.	Pseudo-female
10	E0500	Mam zamiar rozprawić się jutro z tym łajdakiem, przez którego straciłem cały dorobek mojego życia.	Original
11	E0459	Wolał pozostać uczciwym robotnikiem bez kasy niż wspinać się po szczeblach kariery oszukując ludzi.	Pseudo-male
12	E0481	Dermatolog nie dał mi gwarancji, że ten nowy krem nie spowoduje wysypki na mojej skórze.	Pseudo-female
13	E0450	Dobrze by było, gdyby przeczytał wszystkie lektury z epoki pozytywizmu.	Original
14	E0470	Męczy mnie, kiedy moja współlokatorka całymi dniami narzeka na wszystko wokół niej.	Pseudo-male
15	E0500	Mam zamiar rozprawić się jutro z tym łajdakiem, przez którego straciłem cały dorobek mojego życia.	Pseudo-female
16	E0440	Herbata to ulubiony napój na śniadanie wśród wielu Polaków.	Pseudo-female
17	E0461	To śmieszne, że twój duży pies przestraszył się tak małej myszki.	Pseudo-female
18	E0470	Męczy mnie, kiedy moja współlokatorka całymi dniami narzeka na wszystko wokół niej.	Pseudo-female
19	E0490	Marek jest bardzo wrażliwym i czułym mężczyzną, który wspaniale opiekuje się swoją rodziną.	Pseudo-female
20	E0480	Mój romantyczny brat powiedział ostatnio swojej dziewczynie, że dla niej jest w stanie postarać się nawet o gwiazdkę z nieba	Original

Test 3

	Filename	*Keyword in different prosodic contexts*	*Intonation pattern*
1	D1787	To nie jest ziemski tylko **niebiański**.	5,.
2	D1788	Czy on jest **niebiański**?	5,?
3	D1759	Przecież tutaj jest **słoninka**!	5,!
4	D1812	To nie jest wyraz dynamit tylko **bajoński**.	5,.
5	D1776	Ten **zgniłek** - na śmietniku - często wytępuje.	2,?
6	D1799	Wyraz **błogosławieństwo** - w języku polskim - niewiele znaczy.	2,?
7	D1813	Czy on jest **bajoński**?	5,?
8	D1814	Przecież on jest **bajoński**!	5,!
9	D1760	Wyraz **słoninka** - w języku polskim - niewiele znaczy.	2,?
10	D1815	Wyraz **bajoński** - dla ekonomisty - oznacza plajtę.	2,?
11	D1798	Przecież tutaj jest napisane **błogosławieństwo**!	5,!
12	D1786	To nie jest wyraz **niebiański** tylko ziemski.	2,?
13	D1797	Czy dał **błogosławieństwo**?	5,?
14	D1758	Czy tutaj jest **słoninka**?	5,?
15	D1774	Czy tutaj jest **zgniłek**?	5,?
16	D1789	Przecież on jest **niebiański**!	5,!
17	D1775	Przecież tutaj jest **zgniłek**!	5,!
18	D1796	To nie są banały tylko **błogosławieństwo**	5,.
19	D1757	To nie jest wyraz dynamit tylko **słoninka**.	5,.
20	D1773	To nie jest wyraz dynamit tylko **zgniłek**.	5,.

Appendix E Answer sheets

Test 1

Zadanie 1

Instrukcja: Za chwilę usłyszysz 20 zdań. Twoim zadaniem jest wysłuchanie zdań i zapisanie tego, co usłyszysz. Po każdym zdaniu będziesz miał(a) kilka sekund, aby zapisać zdanie.

Numer	*Zdanie*
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

Test 2

Zadanie 2

Instrukcja: Za chwilę usłyszysz 20 długich zdań. Twoim zadaniem jest ocena jakości mowy. Po każdym zdaniu będzie kilkusekundowa przerwa – to czas dla Ciebie, abyś przyznał(a) zdaniu jedną z pięciu ocen: Doskonały – Dobry – Dostateczny – Słaby – Zły.

Numer	*Doskonały*	*Dobry*	*Dostateczny*	*Słaby*	*Zły*
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

Test 3

Zadanie 3

Instrukcje: Za chwilę usłyszysz 20 wyrazów. Twoim zadaniem jest ocena melodii wyrazów. Melodia wyrazów może sugerować, że wyraz ten mógłby pojawić się na końcu zdania oznajmującego lub zdania pytającego. Wyrazy, dla których nie możesz się zdecydować, oznacz "Nie wiem".

Numer	*Zdanie oznajmujące*	*Zdanie pytające*	*Nie wiem*
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

Appendix F Evaluation test results

Subjects

	IP	*Gender*	*Age*	*Nationality*	*Computer knowledge*	*Notes*
1	A-DB	Male	27	Polish	Very good	
2	B-KR	Male	24	Polish	Good	
3	C-NB	Male	15	Polish	Good	Test 1: Listened to the stimuli twice before wrote something.
4	D-DB	Male	39	Polish	Fair	
5	E-PR	Male	35	Polish	Poor	
6	F-ML	Male	14	Polish	Good	Test 1: Listened to the stimuli twice before wrote something, problems memorising the sentences. Did the tasks feverishly.
7	G-KB	Male	20	Polish	Very good	
8	H-MB	Male	55	Polish	Good	Test 1: Listened to the stimuli twice before wrote something.
9	I-JN	Female	9	Polish	Poor	Test 1: Listened to the stimuli twice before wrote something, problems memorising the sentences.
10	J-KM	Female	16	Polish	Poor	
11	K-MM	Female	17	Polish	Fair	
12	L-SW	Female	17	Polish	Very good	
13	M-AB	Female	8	Polish	Fair	
14	N-LB	Female	50	Polish	Fair	Test 1: Listened to the stimuli twice before wrote something
15	O-AM	Female	24	Polish	Very good	

	IP	*Gender*	*Age*	*Nationality*	*Computer knowledge*	*Notes*
16	P-MB	Female	14	Polish	Good	
17	R-EK	Female	23	Polish	Very good	Test 1: Problems memorising sentences.
18	S-AB	Female	36	Polish	Fair	
19	W-JP	Female	22	Polish	Very good	Choir singer
20	T-AS	Female	22	German	Very good	
21	U-PH	Male	25	US-American	Very good	Studying Polish for 7 years.

Test 1

IP	*Unit*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*Sentences (absolute)*	*Words (absolute)*	*Sentences (%)*	*Words (%)*	*Predictable (%)*	*Unpredictable (%)*
M-AB	Sentences	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	0	1	1	15		75		90	60
	Words	4	6	5	4	8	7	6	2	6	4	5	11	5	13	4	3	5	4	6	4		112		89	95	80
A-DB	Sentences	0	1	1	1	1	1	1	0	1	0	1	1	1	1	0	1	1	1	1	1	16		80		100	60
	Words	1	6	5	4	8	7	10	2	6	3	5	11	5	13	2	5	5	6	6	4		114		90	100	76
G-KB	Sentences	1	1	1	1	1	1	1	0	1	0	0	1	1	1	0	1	1	1	1	1	16		80		90	70
	Words	5	6	5	4	8	7	10	3	6	3	4	11	5	13	3	5	5	6	6	4		119		94	99	88
B-KR	Sentences	1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	17		85		100	70
	Words	5	6	2	4	8	7	10	4	6	3	5	11	5	13	5	5	5	6	6	4		120		95	100	88
C-NB	Sentences	0	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	9		45		60	30
	Words	2	5	2	4	8	6	9	2	5	2	5	11	5	11	1	1	5	6	6	4		100		79	93	59
D-DB	Sentences	1	0	1	1	0	1	1	0	1	0	1	1	1	1	0	0	1	1	1	1	14		70		80	60
	Words	5	5	5	4	7	7	10	2	6	2	5	11	5	13	3	2	5	6	6	4		113		90	97	78
E-PR	Sentences	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	17		85		100	70
	Words	5	6	3	4	8	7	10	4	6	4	5	11	5	13	2	5	5	6	6	4		119		94	100	86

IP	*Unit*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*Sentences (absolute)*	*Words (absolute)*	*Sentences (%)*	*Words (%)*	*Predictable (%)*	*Unpredictable (%)*
F-ML	Sentences	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	1	7		35		40	30
	Words	0	4	4	4	6	7	3	2	3	2	5	10	5	12	1	1	5	6	5	4		89		71	81	55
H-MB	Sentences	0	1	1	1	1	1	0	0	1	0	1	1	1	1	0	1	1	0	1	0	13		65		90	40
	Words	4	6	5	4	8	7	6	4	6	3	5	11	5	13	3	5	5	5	6	3		114		90	95	84
I-JN	Sentences	0	1	0	1	1	0	0	0	1	0	1	1	1	1	0	0	1	1	1	0	11		55		80	30
	Words	3	6	4	4	8	6	6	3	6	2	5	11	5	13	4	4	5	6	6	3		110		87	93	78
J-KM	Sentences	0	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	1	0	15		75		90	60
	Words	3	6	5	4	8	7	10	4	6	3	5	11	5	12	5	5	5	6	6	3		119		94	99	88
K-MM	Sentences	1	1	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1	1	1	16		80		90	70
	Words	5	6	5	4	8	7	10	3	5	3	5	11	5	9	5	5	5	6	6	4		117		93	95	90
L-SW	Sentences	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	17		85		100	70
	Words	4	6	5	4	8	7	10	4	6	4	5	11	5	13	4	5	5	6	6	4		122		97	100	92
N-LB	Sentences	0	0	1	1	1	1	1	0	1	0	1	1	1	1	0	0	1	1	1	1	14		70		90	50
	Words	2	4	5	4	8	7	10	2	6	3	5	11	5	13	0	2	5	6	6	4		108		86	97	69
O-AM	Sentences	1	0	1	1	1	0	1	0	0	0	1	1	1	1	0	0	1	1	1	1	13		65		80	50
	Words	5	5	5	4	8	6	10	3	5	3	5	11	5	13	4	2	5	6	6	4		115		91	97	82

IP	*Unit*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*Sentences (absolute)*	*Words (absolute)*	*Sentences (%)*	*Words (%)*	*Predictable (%)*	*Unpredictable (%)*
P-MB	Sentences	0	1	1	1	1	1	1	0	1	0	1	0	1	1	0	0	1	1	1	1	14		70		90	50
	Words	4	6	5	4	8	7	10	4	6	3	5	10	5	13	3	4	5	6	6	4		118		94	99	86
R-EK	Sentences	1	0	0	1	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	11		55		70	40
	Words	5	5	4	4	8	7	7	2	5	3	5	11	5	11	4	5	5	6	6	3		111		88	92	82
S-AB	Sentences	1	0	1	1	1	1	1	0	1	0	1	1	1	1	0	1	1	1	1	1	16		80		90	70
	Words	5	5	5	4	8	7	10	4	6	3	5	11	5	13	4	5	5	6	6	4		121		96	99	92
W-JP	Sentences	1	1	1	1	1	1	0	0	1	0	1	1	1	1	0	1	1	1	1	1	16		80		90	70
	Words	5	6	5	4	8	7	9	4	6	3	5	11	5	13	4	5	5	6	6	4		121		96	99	92

T-AS	Sentences	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3		15		20	10
	Words	3	2	1	4	5	7	6	1	4	1	4	3	3	8	3	2	4	2	4	4		71		56	63	47
U-PH	Sentences	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		5		10	0
	Words	2	3	3	4	3	1	3	1	2	2	3	6	3	2	2	2	4	1	3	1		51		40	43	37

Test 2

IP	*Voice*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*Average*	*STDV*
M-AB	Original	5						5			5			5							5	5	0
	Pseudo -female		3				2		1	4			2			4	4	2	4	2		2,8	1,14
	Pseudo-male			2	4	1						2			1							2	1,22
A-DB	Original	5						5			5			5							5	5	0
	Pseudo -female		3				4		4	4			3			4	4	4	3	4		3,7	0,48
	Pseudo-male			3	2	3						3			2							2,6	0,55
G-KB	Original	4						4			5			5							5	4,6	0,55
	Pseudo -female		2				3			2			3			3	3	3	3	3		2,8	0,44
	Pseudo-male			2	2	2						2			2							2	0
B-KR	Original	5						5			5			5							5	5	0
	Pseudo -female		2				4		3	4			4			3	3	3	4	4		3,4	0,7
	Pseudo-male			2	3	4						3			2							2,8	0,84
C-NB	Original	5						4			5			4							5	4,6	0,55
	Pseudo -female		3				3		3	4			3			3	2	3	2	2		2,8	0,63
	Pseudo-male			1	2	3						3			2							2,2	0,84
D-DB	Original	4						5			5			4							5	4,6	0,55
	Pseudo -female		3				1		3	4			3			3	3	3	2	2		2,7	0,82
	Pseudo-male			2	1	2						4			2							2,2	1,1

IP	*Voice*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*Average*	*STDV*
E-PR	Original	4						5			5			5							5	4,8	0,45
	Pseudo -female		2				4		3	3			2			3	2	2	3	3		2,7	0,67
	Pseudo-male			2	2	3						2			3							2,4	0,55
F-ML	Original	4						5			4			3							5	4,2	0,84
	Pseudo -female		3				2		1	4			2			2	2	3	1	3		2,3	0,95
	Pseudo-male			2	2	4						2			1							2,2	1,1
H-MB	Original	4						5			5			5							5	4,8	0,45
	Pseudo -female		3				4		4	4			4			4	4	4	4	4		3,9	0,32
	Pseudo-male			3	3	3						3			3							3	0
I-JN	Original	5						4			5			4							5	4,6	0,55
	Pseudo -female		1				1		2	2			2			1	3	1	2	3		1,8	0,79
	Pseudo-male			2	2	1						3			2							2	0,71
J-KM	Original	5						5			4			5							5	4,8	0,45
	Pseudo -female		3				3		3	3			3			2	1	2	2	2		2,4	0,7
	Pseudo-male			2	2	2						2			3							2,2	0,45
K-MM	Original	5						5			5			5							5	5	0
	Pseudo -female		4				4		3	4			4			3	3	3	3	3		3,4	0,52
	Pseudo-male			4	3	3						3			3							3,2	0,45
L-SW	Original	4						5			5			4							5	4,6	0,55
	Pseudo -female		1				3		3	2			3			2	1	3	2	3		2,3	0,82
	Pseudo-male			1	2	2						2			1							1,6	0,55

IP	*Voice*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*Average*	*STDV*
N-LB	Original	4						4			5			5							5	4,6	0,55
	Pseudo -female		3				3		3	4			2			2	2	3	2	3		2,7	0,67
	Pseudo-male			2	2	2						2			2							2	0
O-AM	Original	4						4			4			5							4	4,2	0,45
	Pseudo -female		2				3		2	3			2			2	2	3	3	3		2,5	0,53
	Pseudo-male			2	2	1						2			2							1,8	0,45
P-MB	Original	4						5			5			5							4	4,6	0,55
	Pseudo -female		2				3		1	2			3			1	2	3	2	2		2,2	0,45
	Pseudo-male			1	2	2						2			2							1,8	0,45
R-EK	Pseudo -female	5						5			5			5							5	5	0
	Pseudo-male		3				3		3	4			3			2	3	3	3	3		3	0,47
	Male			3	4	4						2			3							3,2	0,84
S-AB	Original	5						4			3			4							5	4,2	0,84
	Pseudo -female		2				1		2	2			2			1	1	1	1	1		1,4	0,52
	Pseudo-male			1	1	1						2			1							1,2	0,45
W-JP	Original	5						5			5			5							5	5	0
	Pseudo -female		3				2		2	3			3			3	4	4	2	3		2,9	0,74
	Pseudo-male			4	3	4						3			2							3,2	0,84

T-AS	Original	4						5			3			4							5	4,2	0,84
	Pseudo -female		3				3		3	2			3			2	2	3	2	4		2,7	0,67
	Pseudo-male			1	1	1						2			1							1,2	0,45

IP	*Voice*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*Average*	*STDV*
U-PH	Original	4						5			5			5							5	4,8	0,45
	Pseudo -female		1				3		2	2			1			1	2	2	2	3		1,8	0,75
	Pseudo-male			2	1	2						1			1							1,4	0,55

Test 3

St – Statement, Q – Question, Ex – Exclamation, ContPh – Continuation phrase.

	Judgement \Input	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	***St***	***Q***	***Ex***	***ContPh***
M-AB	Statement	1		1	1	1	1				1	1	1					1	1	1	1	5	0	3	4
	Question		1					1	1					1	1	1						0	5	1	0
	Don't know									1							1					0	0	1	1
A-DB	Statement			1	1	1	1		1	1	1	1	1				1	1	1	1	1	4	0	5	5
	Question							1						1	1	1						0	4	0	0
	Don't know	1	1																			1	1	0	0
G-KB	Statement			1		1	1					1					1	1	1	1	1	3	0	4	2
	Question		1					1	1		1			1	1	1						0	5	1	1
	Don't know	1			1					1			1									2	0	0	2
B-KR	Statement			1	1	1			1			1						1	1	1	1	4	0	4	1
	Question							1						1	1	1						0	4	0	0
	Don't know	1	1				1			1	1		1				1					1	1	1	4
C-NB	Statement			1	1	1	1					1	1				1	1	1	1	1	4	0	4	3
	Question							1	1		1			1	1	1						0	4	1	1
	Don't know	1	1							1												1	1	0	1

	Judgement \Input	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	***St***	***Q***	***Ex***	***ContPh***
D-DB	Statement			1	1	1	1		1	1		1	1				1	1	1	1	1	4	0	5	4
	Question							1						1	1	1						0	4	0	0
	Don't know	1	1								1											1	1	0	1
E-PR	Statement	1		1	1	1	1		1		1	1	1				1	1	1	1	1	5	0	5	4
	Question		1					1		1				1	1	1						0	5	0	1
	Don't know																					0	0	0	0
F-ML	Statement			1	1	1	1	1	1	1			1			1	1	1	1	1	1	4	2	4	4
	Question		1									1		1	1							0	3	1	0
	Don't know	1									1											1	0	0	1
H-MB	Statement			1	1	1	1		1		1	1	1				1	1	1	1	1	4	0	5	4
	Question		1					1		1				1	1	1						0	5	0	1
	Don't know	1																				1	0	0	0
I-JN	Statement	1		1	1		1					1	1					1	1	1	1	5	0	3	2
	Question		1								1			1	1		1					0	3	1	1
	Don't know					1		1	1	1						1						0	2	1	2
J-KM	Statement			1	1	1	1		1	1	1	1	1				1	1	1	1	1	4	0	5	5
	Question		1					1						1	1							0	4	0	0
	Don't know	1														1						1	1	0	0
K-MM	Statement	1		1	1	1	1		1	1		1	1				1	1	1	1	1	5	0	5	4
	Question							1						1	1	1						0	4	0	0
	Don't know		1								1											0	1	0	1

	Judgement \Input	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	***St***	***Q***	***Ex***	***ContPh***
L-SW	Statement	1				1	1		1			1				1						1	1	2	2
	Question		1					1			1			1	1			1	1	1	1	3	4	1	1
	Don't know			1	1					1			1				1					1	0	2	2
N-LB	Statement	1	1	1	1	1	1		1	1		1	1				1	1	1	1	1	5	1	5	4
	Question							1			1			1	1	1						0	4	0	1
	Don't know																					0	0	0	0
P-MB	Statement	1			1	1					1								1		1	4	0	0	2
	Question		1	1			1	1					1	1	1	1		1				0	5	2	2
	Don't know								1	1		1					1			1		1	0	3	1
R-EK	Statement	1		1	1	1	1		1	1	1	1	1				1	1	1	1	1	5	0	5	5
	Question		1					1						1	1	1						0	5	0	0
	Don't know																					0	0	0	0
S-AB	Statement	1		1	1	1	1					1	1					1	1	1	1	5	0	3	3
	Question		1					1		1				1	1	1	1					0	5	1	1
	Don't know								1		1											0	0	1	1
W-JP	Statement	1	1		1				1	1	1	1	1				1	1	1	1	1	5	1	4	3
	Question			1		1	1	1						1	1	1						0	4	1	2
	Don't know																					0	0	0	0

	Judgement \Input	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	***St***	***Q***	***Ex***	***ContPh***
T-AS	Statement	1	1	1	1	1				1			1					1	1		1	4	1	2	3
	Question							1	1		1	1		1	1	1						0	4	2	1
	Don't know						1										1			1		1	0	1	1
U-PH	Statement							1						1	1	1						0	4	0	0
	Question		1	1		1			1			1	1				1	1	1		1	2	1	5	2
	Don't know	1			1		1			1	1									1		3	0	0	3