Advances in Speech Recognition

edited by
Noam R. Shabtai

SCIYO

Advances in Speech Recognition
Edited by Noam R. Shabtai

Published by Sciyo
Janeza Trdine 9, 51000 Rijeka, Croatia

Publishing Process Manager Jelena Marusic
Technical Editor Teodora Smiljanic
Cover Designer Martina Sirotic
Image Copyright Aleksandar Zoric, 2010. Used under license from Shutterstock.com

First published September 2010
Printed in India

A free online edition of this book is available at **www.sciyo.com**
Additional hard copies can be obtained from **publication@sciyo.com**

Advances in Speech Recognition, Edited by Noam R. Shabtai
p. cm.
ISBN 978-953-307-097-1

Contents

Preface VII

Chapter 1 **Extra Large Vocabulary Continuous Speech Recognition Algorithm based on Information Retrieval 1**
Valeriy Pylypenko

Chapter 2 **Neuro-Inspired Speech Recognition Based on Reservoir Computing 7**
A Ghani, T.M. McGinnity, L Maguire, L McDaid and A Belatreche

Chapter 3 **The Effect of Reverberation on Optimal GMM Order and CMS Performance in Speaker Verification Systems 37**
Noam R. Shabtai, Boaz Rafaely and Yaniv Zigel

Chapter 4 **Body-Conducted Speech Recognition and its Application to Speech Support System 51**
Shunsuke Ishimitsu

Chapter 5 **Modelling of Filled Pauses and Onomatopoeias for Spontaneous Speech Recognition 67**
Andrej Žgank and Mirjam Sepesy Maučec

Chapter 6 **Non-native Pronunciation Variation Modeling for Automatic Speech Recognition 83**
Mina Kim, Yoo Rhee Oh and Hong Kook Kim

Chapter 7 **Applications of Speech Technologies in Western Balkan Countries 105**
Darko Pekar, Dragiša Mišković, Dragan Knežević, Nataša Vujnović Sedlar, Milan Sečujski and Vlado Delić

Chapter 8 **Croatian Speech Recognition 123**
Ivo Ipšić and Sanda Martinčić–Ipšić

Chapter 9 **Speech Technologies for Serbian and Kindred South Slavic Languages 141**
Vlado Delić, Milan Sečujski, Nikša Jakovljević, Marko Janev, Radovan Obradović and Darko Pekar

Preface

Speech processing has come a long way since the year of 1947, when R. K. Potter, G. A. Kopp, and H. Green from Bell Labs introduced the sound spectrograph, the first instrument to produce human voice-prints in the short-time Fourier-transform domain. Ever since, speech recognition has been constantly evolving. From isolated word recognition with small vocabulary in the 1950s and medium vocabulary in the 1960s, speech recognition advanced through connected words recognition with large vocabulary in the 1970s and 1980s, to very large vocabulary continuous speech recognition in the 1990s and 2000s.

During the 1950s, speaker recognition systems were capable of recognizing only 10 isolated words or digits. The applications were speaker dependent, meaning that systems were capable to recognize word utterances from a single speaker. In 1952, an isolated digits recognition system was developed by K. H. Davis, R. Biddulph, and S. Balashek from Bell Labs. Independently in 1956, H. F. Olson and H. Belar from the RCA laboratories have developed a speech recognition system capable of recognizing 10 isolated monosyllabic words.

Japanese vowel recognition was performed in 1960 by J. Suzuki and K. Nakata from the Radio Research Lab in Japan. The IBM Shoe-Box computer, introduced in 1962, was capable of performing 16 words and digits recognition. In the same year, the first phoneme recognizing hardware was built in Kyoto University in Japan by T. Sakai and S. Doshita. In 1966, J. L. Flanagan and R. M. Golden have presented the phase vocoder. Although created for the purpose of speech coding rather than speech recognition, it supplied a great deal of insight into the short-time processing of speech signals. Dynamic time warping was first used by T. K. Vintsyuk in 1968 (improved in 1971, and further in 1978, by H. Sakoe and S. Chiba). At that time, a 54 isolated-word speech recognition system, shown by D. G. Bobrow and D. H. Klatt, was the state-of-the-art. The first continuous speech recognition system appeared in 1969, developed by D. R. Reddy from Carnegie Mellon University. In 1969, B. P. Bogert, M. J. R. Healy, and J. W. Tukey have minted the concept of cepstrum, and introduced the cepstral analysis.

In 1970, F. Itakura and S. Saito presented the *linear predictive coding* (LPC) method for speech spectrum and formant estimation, leading the way to more effective speech feature extraction methods. The LPC feature vectors have become a very important tool in speech recognition and in speech signal processing in general. The speech recognition systems in 1972 were capable of recognizing 100 words, and by 1974, 200 words. The Itakura-distance measure was defined for LPC feature vectors in 1975 by Itakura, who was then working at Bell Labs. In the same year, *hidden Markov models* (HMMs), which were previously introduced by L. E. Baum in a series of statistical papers in the 1960s, were implemented in the DRAGON speech recognition system by J. K. Baker. This development marked the replacement of template-based approaches to speech recognition by statistically-based methods. In 1977, 30 years after the appearance of the sound spectrograph, J. B. Allen and L. R. Rabiner published a unified approach to short-time Fourier analysis and synthesis.

In 1980, S. D. Davis and P. Mermelstein introduced the mel-frequency cepstral coefficients (MFCCs), which improved the cepstral analysis by using known characteristics of the human auditory system. The MFCC feature vectors constitute a major tool in speech feature extraction, and an alternative to the LPC feature vectors. In 1980 IBM presented the first small messages dictating machine. Time-varying parametric modeling of speech was introduced in 1982 by M. G. Hall, A. V. Oppenheim, and A. S. Willsky. During the 1980s, HMM has become a dominant approach in speech recognition, rather than implemented only by a few developers (such as IBM, IDA, and DRAGON). This was mainly due to the works published by Rabiner, including the HMM tutorial appearing in the proceedings of the IEEE, published in 1989.

In 1990, the *perceptual linear predictive* (PLP) speech processing technique was introduced by H. Hermansky, and improved in 1994 by further introducing the relative spectral methodology (RASTA), to form the RASTA-PLP feature vectors. During the 1990s and the 2000s, state-of-the-art speech recognition systems were using evolved HMM variants, human perceptual versions of cepstral or linear predictive coding feature vectors, and sophisticated pattern-matching and scoring algorithms.

In the last decade, further applications of speech processing were developed, such as speaker recognition, human-machine interaction, non-English speech recognition, and non-native English speech recognition. This book addresses a few of these applications. Furthermore, major challenges that were typically ignored in previous speech recognition research, such as noise and reverberation, appear repeatedly in recent papers. The first part of the book, *applications in speech recognition*, begins with a development of large vocabulary continuous speech recognition algorithm. It continues with several speech recognition innovations, and various applications in speech processing, such as speaker recognition (the relevant chapter also deals with reverberation challenges), and body-conducted speech recognition. The second part of the book, *non-native and non-English speech recognition*, introduces speech recognition of English spoken by non-native speakers, and speech recognition of non-English languages.

I would like to sincerely thank the contributing authors, for their effort to bring their insights and perspectives on current open questions in speech recognition research. I would also like to express my deepest appreciation and gratitude to the SCIYO organization, and to the editorial office, which gathered the authors and published this book.

Editor

Noam R. Shabtai
Ben-Gurion University of the Negev,
Israel

In memory of Reuven Yechezkel

1

Extra Large Vocabulary Continuous Speech Recognition Algorithm based on Information Retrieval

Valeriy Pylypenko
International Research/Training Center for Information Technologies and Systems
Kyiv, Ukraine

1. Introduction

There exists a necessity for speech recognition with a huge numbers of alternatives.
For example, during the voice control of a computer it is impossible to predict the subsequent word on the basis of several previous ones because it is defined by control logic, instead of text properties. From the other hand there is a necessity for growth of the volume of the dictionary to capture all possible synonyms of the same command caused by difficulty for users to remember the single command name variant.
The next example concerns the text dictation. The application of such systems is limited by the texts, which are statistically similar to one where statistics were collected. Additional spoken editing of the text demands the presence of all words in the actual dictionary.
Thus, there are applications where it is desirable to have a dictionary as large as possible, in future to cover all words for the given language (for some languages more than 10M words).
The additional information to restrict the number of alternatives can be received from a speech signal immediately. For this purpose it is proposed to execute preliminary trial recognition by using the phonetic transcriber. Phonemes sequence analysis allows to build the queries flow. Applying the information retrieval approach considerably limits the number of alternatives for recognition.

2. The baseline recognition systems

The approach is applicable for any recognition system where phonemes and phoneme recognition (phonetic transcriber) are present but the number of phonemes no more than approximately 500 units.
As reference systems HMM-based HTK (Young et al., 2006) and Julius (Lee, 2009) toolkits are used.

3. ELVIRS Algorithm for isolated words

3.1 Architecture

The architecture of the system is shown in Figure 1. The *features extraction* and *acoustic models* blocks are reused from the baseline system. Common *pattern matching* unit with subset of

vocabulary is used on the second pass. Changes are concentrated in the new first recognition pass when *phonetic transcriber* is applied to make the sequence of phonemes. Then *information retrieval procedure* builds the sub-vocabulary for the second pass.

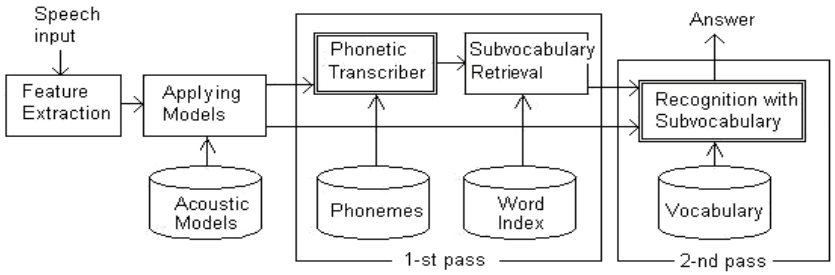


Fig. 1. The architecture of ELVIRS recognition system

3.2 Phoneme recognizer

The phonetic transcribing algorithm (Vintsiuk, 2000; Vintsiuk, 2001) builds a phonetic sequence for speech signal regardless to the dictionary. For this purpose a phoneme generative automaton was constructed which can synthesize all possible continuous speech model signals for any phoneme sequence. Then the phoneme-by-phoneme recognition of unknown speech signal is applied.

The same context-free phonemes as in baseline recognition system are used.

The experimental accuracy of finding phoneme at the right place equals to approximately 85%.

3.3 Sub-vocabulary retrieval procedure

Preliminary transcription dictionary is prepared to build phoneme triples. The index entry key is a phoneme triple, thus, the index consists of M^3 entries where M is the number of phonemes in the system. Each index entry contains the list of transcriptions that include key phoneme triple. Additional memory usage is approximately 50 MB for vocabulary with 1 M words.

Sub-vocabulary retrieval process is illustrated in Figure 2. Phoneme recognizer output is split into overlapping phoneme triples. Resulting phoneme triple becomes the query. Now, in this system the simple query is used where phoneme triple and query are the same. In the future it should be modified to take into account the insertion, deletion and substitution of phoneme sequence by using *Levenshtein* dissimilarity. Thus phonetic sequence produces the query flow for database.

The query answer consists of the list of transcriptions in which the given triple is included. Next queries produce new transcription portions to be copied into the sub-vocabulary for the second pass. The counter for word repetition is supported to make the rank of word.

All transcriptions in resulting sub-vocabulary are arranged according to the word rank (repetition counter). First N transcriptions are copied into a final sub-vocabulary for the second pass recognition. Thus the recognition sub-vocabulary consists of transcriptions of highest ranks but the vocabulary size does not exceed a fixed limit N.

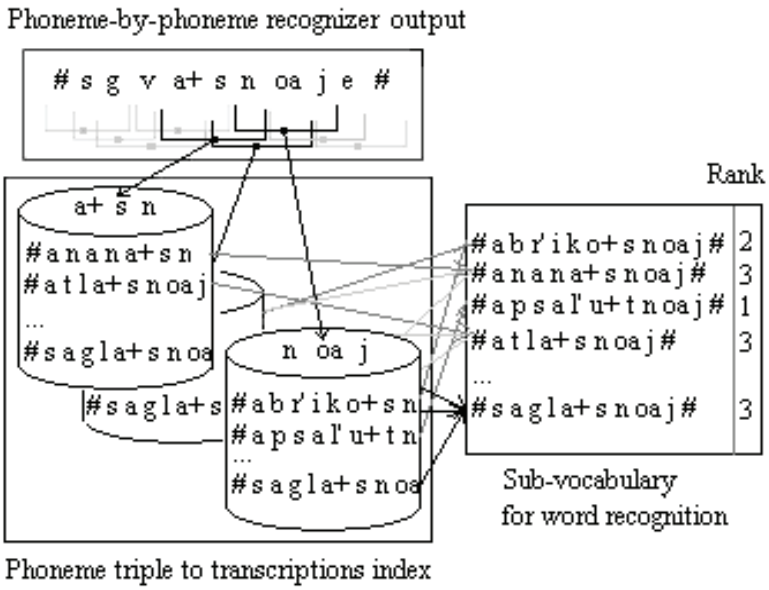


Fig. 2. Sub-vocabulary retrieval process

3.4 ELVIRS algorithm overview

The ELVIRS algorithm (Pylypenko, 2006) works as described in the following.

Preparation stage:

1. Prepare the recognition vocabulary.
2. Chose the phoneme set and build transcriptions for words from vocabulary by rules.
3. Create database index from phoneme triple to transcriptions.
4. Train the acoustic models from collected speech signals.

Recognition stage:

1. Apply phoneme recognizer for input speech signal to produce a phoneme sequence.
2. Split the phoneme sequence into overlapping phoneme triples.
3. Make queries from phoneme triples.
4. Retrieve transcription lists by queries from database index.
5. Arrange transcriptions by the rank.
6. Chose N-best transcriptions for recognition sub-vocabulary.
7. Recognize the input speech signal with sub-vocabulary.

4. The information consideration

The phoneme recognizer output can be considered as a correct phoneme sequence passed through a noisy channel and converted into an output sequence. Denote a right phoneme in output sequence as 1 and wrong one as 0. Let 1 occurs with probability u. The probability P to find k and more successive 1 in a binary set with length of n can be computed with the help of the following recurrent expression:

$$P_n = \begin{cases} 0, n < k \\ u^k, n = k \\ P_{n-1} + u^k(1-u)(1-P_{n-k-1}), n > k \end{cases}$$

Probabilities P to find three and more successive 1 in a binary sequence for different lengths n and probabilities u are shown in Table 3. Average transcriptions length is equal to approximately 8 and the accuracy of finding phoneme at the right place for known utterance is equal to approximately 85%. For these values the probability to find right word in chosen sub-vocabulary is equal 0.953

n \ u	0.75	0.8	0.85	0.9
6	0.738	0.819	0.890	0.948
7	0.799	0.869	0.926	0.967
8	0.849	0.908	**0.953**	0.982
9	0.887	0.937	0.971	0.991
10	0.915	0.956	0.981	0.995

Table 1. Probability to find three and more successive 1 in a binary sequence with length of n

5. ELVIRCOS algorithm for continuous speech

5.1 Architecture

After transcriptions list retrieval procedure an additional procedure – word graph composition is applied. It produces a word network for second pass recognition.

5.2 Word graph composition

The word graph composition procedure is illustrated in Figure 3. Word network starts from vertex S and ends at vertex F. Each triple from phoneme output burns intermediate vertexes with numbers synchronous the occurrence time. On the other hand, each triple became query to data base index, which returns the transcription list as result. Transcriptions are interlaced with intermediate numbered vertexes as base vertexes so that burning phoneme triples are placed in coordination.

The rank of transcription is increased in case when intersection between same transcriptions burned from different phoneme triple occurs. For each moment of time (synchronous with phoneme sequence) the number of involved transcriptions may be calculated.

In order to reduce the word graph complexity, the fixed limit N is applied. For each moment of time transcriptions with small ranks are removed from word graph so that only N transcriptions remain.

The word graph is composed from left to right, that is why it is possible to construct one in real time with the delay is equal of largest transcription length.

5.3 ELVIRCOS algorithm overview

The ELVIRCOS algorithm (Pylypenko, 2007) works as follows.

Preparation stage is the same as ELVIRS algorithm.

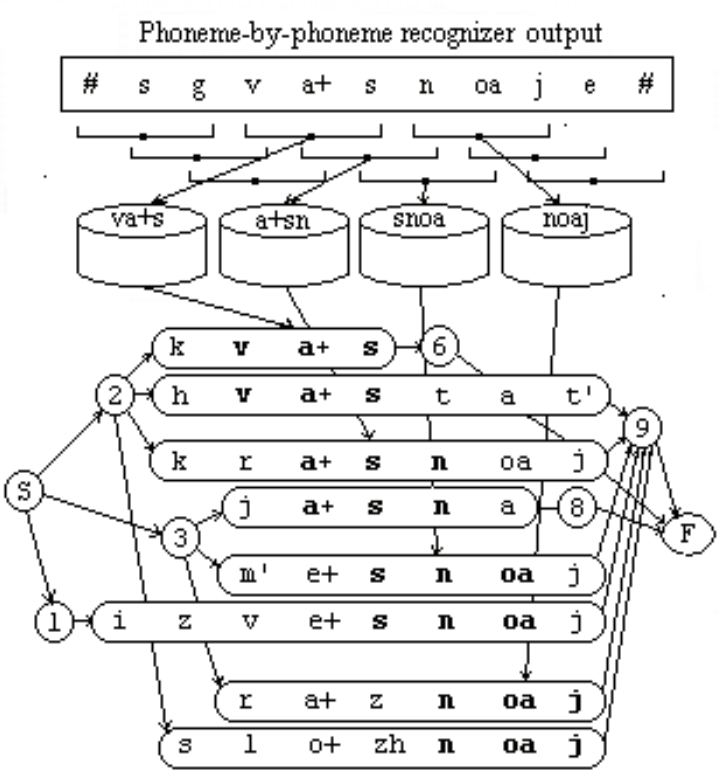


Fig. 3. Word graph composition for continuous speech

Recognition stage:

1. Apply phoneme recognizer to the input speech signal to produce a phoneme sequence.
2. Split the phoneme sequence into overlapping phoneme triples.
3. Make queries from phoneme triples.
4. Retrieve transcription lists by queries from database index.
5. Compose word graph network.
6. Recognize the input speech signal with composed word net.

6. Experimental results

The algorithm was tested at speech corpus from 3 sources:

1. Russian isolated and continuous speech from one speaker with duration 2 hours for training and 20 min for testing.
2. Ukrainian Parliament speech from about 200 speakers with duration 50 hours for training and 3 hours for testing.
3. The November 1992 ARPA Continuous Speech Recognition Wall Street Journal Benchmark Tests.

For experiments, some modifications of HTK or Julius toolkit were necessary to take into account the algorithm.

The considerable reduction of the recognition time (about 10-50 times) with relatively small accuracy degradation (approximately 5%) in comparison with baseline systems has been achieved. The accuracy degradation has a good agreement with the information consideration.
Recognition time not depends from vocabulary size but requires some enlarging because the recognition accuracy fall with vocabulary growth and needs to pay compensation by taking in to account more amount of hypothesis.

7. Future extension

The importance of information retrieval for speech recognition should be underlined. It was shown that additional information source from analysis of phoneme sequence allows to restrict the search space. These new restrictions lead to speech recognition with vocabularies cover practically all words for given language.
Now some modification to adopt bigram language model is developing as a new direction for proposed algorithm. More complex language models can be applied in future works to achieve new features.

8. References

Lee, A., "The Julius book", http://julius.sourceforge.jp, 2009

Pylypenko, V. (2006). "Information Retrieval Based Algorithm for Extra Large Vocabulary Speech Recognition", *Proc. of the 11th International Conference "Speech and Computer", SPECOM'2006*, St. Petersburg, Russia, 2006.

Pylypenko, V. (2007) "Extra Large Vocabulary Continuous Speech Recognition Algorithm based on Information Retrieval", *Proc. of the 8th International Conference "Interspeech 2007"*, Antwerp, Belgium, 2007.

Vintsiuk, T. K. (2000) "Generalized Automatic Phonetic Transcribing of Speech Signals", *Proc. of the 5th All-Ukrainian Conference "Signal/Image Processing and Pattern Recognition"*, pp. 95–98, Kyiv, Ukraine, 2000, in Ukrainian

Vintsiuk, T. K. (2001) "Generative Phoneme-Threephone Model for ASR", *Proc. of the 4th Workshop on Text, Speech, Dialog – TSD'2001*, p. 201, Zelezna Ruda, Czech Republic, 2001.

Young, S.; Kershaw, D.; Odell, J.; Ollason, D.; Valtchev, V. & Woodland, P. (2002). *The HTK Book (for HTK Version 3.2)*. Cambridge University, Cambridge, UK.

2

Neuro-Inspired Speech Recognition Based on Reservoir Computing

A Ghani, T.M. McGinnity, L Maguire, L McDaid and A Belatreche
University of Ulster
N. Ireland, United Kingdom

1. Introduction

This chapter investigates the potential of recurrent spiking neurons for classification problems. It presents a hybrid approach based on the paradigm of *Reservoir Computing*. The practical applications based on recurrent spiking neurons are limited due to the lack of learning algorithms. Most of the previous work in the literature has focused on feed forward networks because computation in these networks is comparatively easy to analyse. The details of such networks have been reported in detail in (Haykin, 1999) (Pavlidis et al., 2005) (Bohte et al., 2000). Recently, a strategy proposed by Maass (Maass et al., 2002) and Jaeger (Jaeger, 2001) offers to overcome the burden of recurrent neural networks training. In this paradigm, instead of training the whole recurrent network only the output layer (known as readout neuron) is trained.

This chapter investigates the potential of recurrent spiking neurons as the basic building blocks for the liquid or so called *reservoir*. These recurrent neural networks are termed as microcircuits which are viewed as basic computational units in cortical computation (Maass et al., 2002). These microcircuits are connected as columns which are linked with other neighboring columns in cortical areas. These columns read out information from each other and serve both as reservoir and readout. The reservoir is modeled as a dynamical system perturbed by the input stream where only readouts are trained to extract information from the reservoir. The basic motivation behind investigating recurrent neurons is their potential to memorise relevant events over short periods of time (Maass et al., 2002). The use of feedback enables recurrent networks to acquire state representation which makes them suitable for temporal based applications such as speech recognition. It is challenging to solve such problems with recurrent networks due to the burden of training. The paradigm of *reservoir computing* also referred to as *liquid computing* relaxes the burden of training because only an output layer is trained instead of training the whole network. The work presented in this chapter analyses the theoretical framework of *Reservoir Computing* and demonstrates results in terms of classification accuracy through the application of speech recognition. The design space for this paradigm is split into three domains; front end, reservoir, and back end. This work contributes to the identification of suitable front and back end processing techniques along with stable reservoir dynamics, which provides a reliable framework for classification related problems.

The work presented in this chapter suggests a simple and efficient biologically plausible approach based on a hybrid implementation of recurrent spiking neurons and classical feed

forward networks for an application of isolated digit recognition. The structure of this chapter is as follows: section 2 elaborates the motivation, related work, theoretical review and description of the paradigm of reservoir computing. Section 3 contains details about the experimental setup and investigates front-end pre-processing techniques and reservoir dynamics. A baseline feed forward classifier is described in section 4 and results are presented. Results based on reservoir recognition are presented in section 5. Section 6 discusses results obtained through Poisson spike encoding. A thorough discussion and conclusion of the chapter is provided in section 7.

2. Computing with recurrent neurons

The paradigm of reservoir or liquid computing is promising because it offers an alternative to the computational power of recurrent neural networks, however analytical study of such networks is not trivial (Legenstein et al., 2003) (Joshi & Maass, 2005) (Jaeger & Haas, 2004). It facilitates training in a recurrent neural network where a linearly non separable low dimensional data is projected on a high dimensional space. The readout of the reservoir can be trained with partial information extracted from the reservoir which suffices to solve complex problems such as speech recognition. In this approach, readout only observes the membrane potential of the spiking neurons at particular time steps which is far more efficient than fully quantifying the reservoir dynamics. It is due to this property that relatively simple readout can be trained with meaningful internal dynamics of the reservoir.
The framework of reservoir computing is more suitable for hardware implementation because network connections remains fixed in the network and there is no need to implement weight adaptation for recurrent reservoirs. This paradigm is inherently noise robust therefore more suitable for digital hardware implementation on reconfigurable platforms such as FPGAs. FPGA implementation of recurrent spiking neurons gives the flexibility to develop such networks with simple real-world interface and offers other desirable features such as noise robustness. This paradigm appears to have great potential for engineering applications such as robotics, speech recognition, and wireless communication (Joshi & Maass, 2004) due to the computationally inexpensive training of readout neurons. This paradigm can also be used for channel equalization of high speed data streams in wireless communication as suggested by (Jaeger & Haas, 2004).
Since the inception of the theoretical foundation by Jaeger and Maass, various groups have focused on investigating different aspects of the paradigm for engineering applications e.g., Skrownski et al., investigated the paradigm of echo state networks for speech recognition applications where the HFCC (Human Factor Cepstral Coefficient) technique was investigated for front end processing and HMM (Hidden Markov Model) classifier was used for back end processing. The overall performance was compared with the baseline HMM classifier. The main focus of the work was to investigate the noise robustness of the system based on echo state networks (Skowronski & Harris, 2007).Verstraeten et al., analysed the classification accuracy of a reservoir with different benchmarks. In their study, both sigmoidal and LIF (Leaky Integrate-and-Fire) based reservoirs were tested for evaluating the memory capacity and overall classification accuracy was calculated based on different sizes of the reservoir. The memory capacity was analysed by evaluating the maximum number of patterns that could be stored for short period of time and memory is analysed by different circuit connections in the reservoir. Moreover, different speech pre processing techniques were also elaborated and their robustness is measured against overall system performance (Verstraeten et al., 2007).

In previous studies, different solutions have been proposed in order to improve the reservoir dynamics to get better accuracies. Maass and Jaeger stated that it is possible to obtain a stable reservoir if topology and weights are drawn randomly (Maass et al., 2002). Jaeger emphasised to control the scaling of the weights while Maass emphasised that stable dynamics can be obtained with proper connection topologies. The objective in both cases was to ensure the property of *fading memory* or *echo state* in the network. It appeared from these studies that the paradigm does not depend on a specific connectivity of a reservoir rather more on a distributed, stable and redundant representation of the neurons. In a recent study, Ismail Uysal investigated a noise robust technique by using phase synchrony coding (Uysal et al., 2007). However, in all of these studies, there are no specific guidelines regarding implementation and stability of reservoirs and all reported techniques that significantly vary from each other based on the authors' experiences of reservoir computing. Implementing stable reservoir is a challenging task, however the stability of the reservoir is not the only criteria which will guarantee a solution to the problems at hand. Two important factors are the proper investigation of front and back end techniques. Regardless of the size of the reservoir or processing nodes, it is rather difficult to solve a problem without investigating a robust front end technique. This work will investigate three main areas: a robust front end, a stable and compact reservoir, and an efficient back end engine for the task of recognition. The overall accuracy of the reservoir based classification technique will be compared with the baseline feedforward network.

2.1 Theoretical background

In contrast to feedforward networks, where inputs are propagated to the output layer in a feedforward manner, feedback loops are built into the design of recurrent networks in order to incorporate dynamical properties. One of the simple and well known architecture was introduced by John J Hopfield (Hopfield, 1982) and Elman in 1990 (Elman, 1990). Other modified architectures based on recurrent neural networks (RNN) have been proposed by Jordan (Jordan, 1996) and Bengio (Bengio, 1996) (see Fig. 2). The Hopfield networks consists of a set of neurons and corresponding unit delays with no hidden units as illustrated in Fig. 1. In this network, the total numbers of feedback loops are equal to the total number of neurons where the output of each neuron is fed back to each of the other neurons in the network with no self feedback loop (Haykin, 1999). Hopfield neural networks are promising but they require large number of neurons compared to the number of classes and take considerably more time to compute compared to feedforward networks (Looney, 1997). Elman network commonly is a two layer network where output from the first layer is fed back to the input of the same layer. A short term memory can be implemented by including delay in the connection which stores values from the previous time step that can be used in the current time step. Because of this short term memory capability, Elman networks can be trained to respond to the spatio-temporal patterns. Jordan type artificial neural networks are recurrent networks with delayed loopback connections between a context output and input layer. The context layer allows the network to produce different values with the same input based on the history. In Bengio networks, the network response is fed back to the input through a context layer and a delayed output from the previous time steps.

Recurrent neural networks have also been investigated by other researchers such as (Doya, 1995) (Atiya, 2000) and (Pearlmutter, 1995). Recurrent neural networks can be represented as a Mealy state machine which receives inputs and produces outputs that are dependent both on its internal state and the input (see Fig. 3).

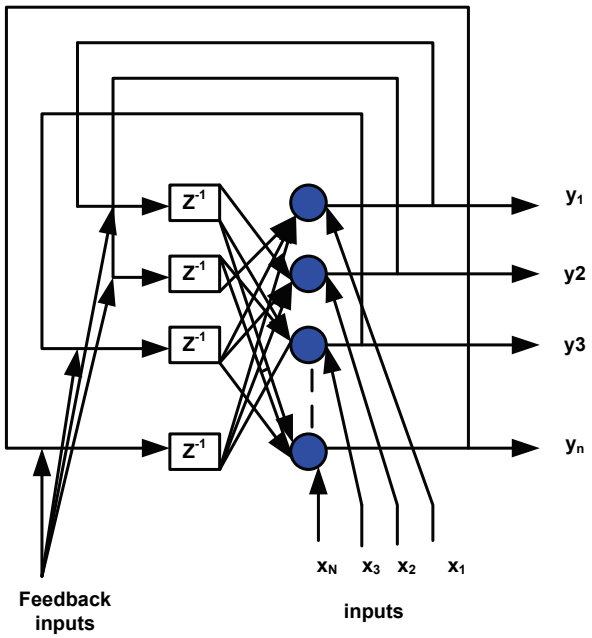


Fig. 1. This figure shows Hopfield network model which consists of a set of neurons and corresponding set of unit delays. This makes this model a recurrent multiple loop feedback system.

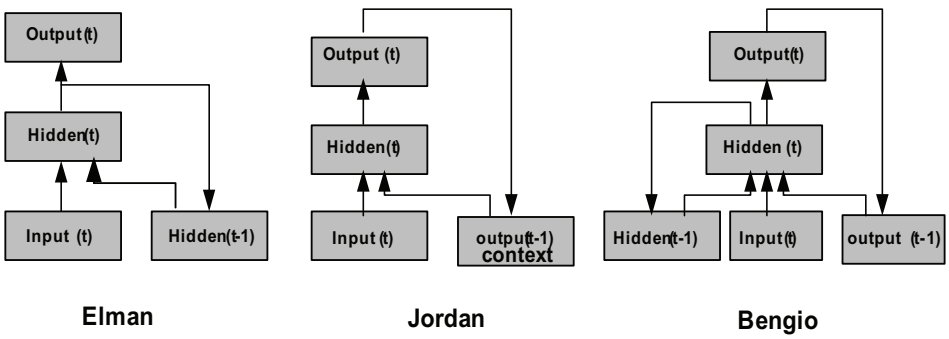


Fig. 2. Recurrent neural networks

A major advantage of recurrent neural networks such as Elman, Bengio, and Jordan is their capability to store information for a short period of time. In feedforward networks, memory can be augmented with tapped delay lines while recurrent networks are provided with build-in memory by recurrent loops (Haykin, 1999). There is no straight forward way to construct a recurrent neural network which will work as a finite state machine, therefore RNNs have to be trained to simulate a specific problem. There has been limited success in this regard and there is no standard algorithm for training RNNs (Jaeger, 2001). There have

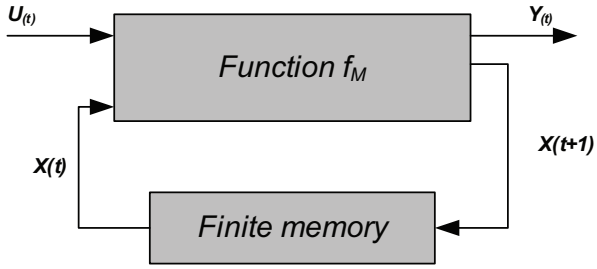


Fig. 3. Mealy type finite state machine where output *Y(t)* is dependent on the input *U(t)*, internal state, *X(t)* and the function f^M.

been some attempts to develop learning algorithms for recurrent networks but they are computationally much more expensive and non-trivial to converge (Haykin, 1991) (Jaeger, 2002) (Williams & Zipser, 1989) (Singhal & Wu, 1989) (Atiya et al., 2000). Some of these algorithms are based on approximating the gradient, others are based on approaches such as extended Kalman filter, EM (expectation-maxmisation) based algorithms and novel architectures such as focused backpropagation and the approximated Levenberg-Marquardt algorithm. A detailed discussion about training of recurrent neural networks is provided by (Atiya et al., 2000). There has been significant research in the temporal phenomena at the synapse level (Abbott & Nelson, 2000) but less emphasis on the learning dynamics at the network level (Jaeger, 2001).

Recently, in order to overcome the burden of training in recurrent networks, the paradigm of liquid computing was introduced by Maass and Jaeger. This paradigm covers three main techniques in classification related problems: Echo State Machine (Jaeger, 2001), Backpropagation Decorrelation (Steil, 2004), and Liquid State Machine (Maass et al., 2002). The fundamental motivation behind all these techniques is to overcome the computational burden of the recurrent neural network training. In the paradigm of liquid computing, the partial response of a recurrent reservoir is observed from outside by any suitable classification algorithm such as back propagation. It is much easier and more computationally efficient to train the output layer (feedforward network) or so called 'readout' neurons, instead of the complete network of recurrent neurons. An abstract overview of the Liquid State Machine is shown in Fig. 4.

The mathematical theory of liquid computing is based on the observation that if a complex recurrent neural circuit is excited by an input stream *u(t)* and after some time *s*, such that when $t > s$ a liquid state *x (t)* is captured, then it is very likely that this state will contain most of the information about recent inputs. According to the theory, it is not possible to understand the neural code but it is not important because liquid by itself serves as short term memory and the major task of learning depends on the state vector *x(t)* which is exclusively used by the *readout* neurons. The liquid reservoir transforms the input stream *u(t)* to a high dimensional spatial state *x(t)* (Maass et al., 2002). The paradigm is shown in Fig. 5.

The liquid state machine (LSM) is somewhat similar to the finite state machine (FSM) but the major difference is that LSM is viewed as a state machine with no limited states in contrast to the FSM where all transitions are custom designed and pre determined. According to the theory, if the reservoir state *x(t)* is high dimensional and if its dynamics are sufficiently complex then many concrete finite state machines are embedded in it. Mathematically, a

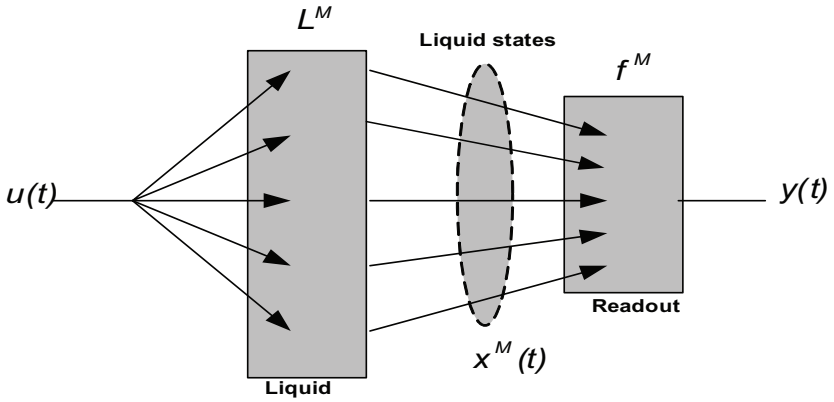


Fig. 4. An abstract overview of a liquid state machine where an input stream $u(t)$ is mapped to a target function $y(t)$. An input is injected to the liquid filter L^M and state vector $x^M(t)$ is captured at each time step t. The state vector is applied to the readout neurons through mapping f^M in order to approximate the target function $y(t)$ (figure annotated from Maass et al., 2002).

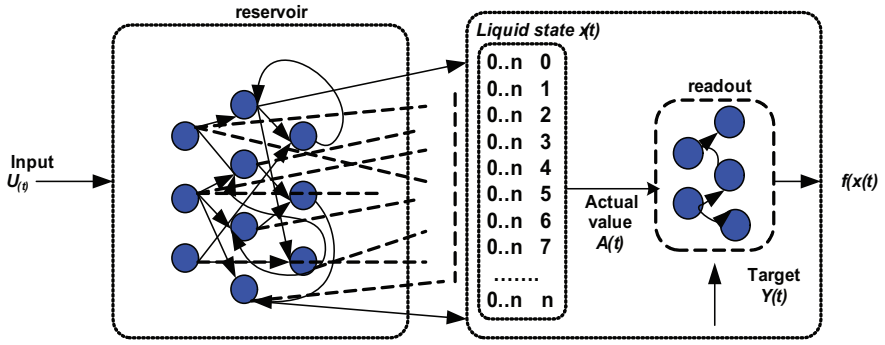


Fig. 5. An abstract overview of a neural reservoir which shows an input $U(t)$ is fed into the spiking recurrent reservoir where different states are captured in the block 'liquid state' which are used as training vectors for readout neurons.

liquid state machine M consists of a filter L^M that maps input stream $u(t)$ onto reservoir state $x(t)$, where $x(t)$ not only depends on $u(t)$ but also on previous input $u(s)$ (Maass et al., 2002). Mathematically this can be written as:

$$x(t) = (L^M u)(t) \tag{1}$$

While a readout function f^M maps the state of the liquid $x(t)$ into a target output $y(t)$.

$$y(t) = f^M (x^M(t)) \tag{2}$$

The advantage of the neural reservoir is that it does not require a task specific connectivity and it does not require any specific code by which information is represented in the neural

reservoir because only the readout neurons are trained (Maass et al., 2002). Theoretical results imply that this is a universal state machine which has no limitation on the power of neural microcircuit as long as the *reservoir* and readouts fulfill the separation and approximation properties. In order to construct a neural microcircuit, the following three steps are required:
The structure of a neural reservoir is defined in terms of processing node types (LIF, HH or Izhikevich), total number of recurrent neurons, their connectivity and parameters. Whereas, the state vector $x(t)$ of the neural reservoir is recorded at different time steps for different inputs $u(t)$. A supervised learning algorithm is applied to train a readout function f such that an actual output $f(x(t))$ is as close as possible to the target value $y(t)$. The simulation experiments performed by Jaeger and Maass showed that a simple readout would be sufficient to extract information from the recurrent neural reservoir. The major difference between ESN and LSM are the node types where sigmoid neurons are used for ESN and LIF neurons for LSM. The results are demonstrated with classification problems as reported in (Jaeger, 2001) (Maass et al., 2002) (Maass et al., 2004b). Maass et al., examined the recurrent LIF neurons in a bench-mark task proposed by (Hopfield & Brody, 2001) (Hopfield & Brody, 2000). The robustness of a neural reservoir is justified by Cover's separability theorem, which states that if a pattern classification problem is projected non-linearly on a high dimensional space, it is more likely to be linearly separable in comparison to the low dimensional space (Cover, 1965) (Skrownski et al., 2007).
It was stated by Maass and Jaeger that temporal integration can be achieved by randomly created recurrent neural reservoirs and various readouts can be trained with the same reservoir. A classification can be guaranteed by this paradigm if the dynamics of a reservoir exhibit a property of *fading memory* or *echo state property*. The concept of echo state property or fading memory is introduced because different states disappear over time. Theoretically, this paradigm appears to have no limits on the power of the neural microcircuit but there are no specific guidelines as how to construct a stable or ordered recurrent neural reservoir, an appropriate front end and classification algorithm for backend readout. In the following section, an experimental framework is proposed inspired by the paradigm of reservoir computing where the design space is split into three main areas: front end, back end and reservoir. Each one of these areas were analysed individually and then integrated for their performance evaluation (Ghani et al., 2006) (Ghani et al., 2008).

3. Experimental setup

The task of isolated spoken digit recognition is significantly complex and different techniques for solving this problem have been reported in the literature (Sivakumar et al., 2000) (Zhao, 1991) (Kim et al., 1999). In this section, a biologically plausible hybrid engine is proposed inspired by the framework of reservoir computing to solve isolated digit recognition. A rather simple feed forward multilayer perceptron is proposed and used as *readout* for classification of 10 isolated spoken digits from the TI46 speech corpus (Doddington & Schalk, 1981). The readout neurons were trained with standard back propagation algorithms and with only partial information extracted from recurrent reservoir being used in the training. It is computationally expensive to process all the states recorded from the recurrent reservoir and therefore inefficient for backend classification. In this experiment, only five states were recorded sampled at every 25 *ms* from start to the end of a 1 second simulation of the reservoir. A detailed design flow of the reservoir classification engine is shown in Fig. 6.

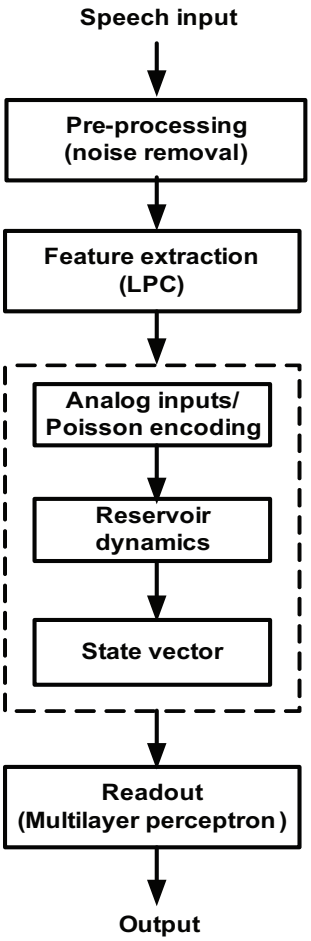


Fig. 6. This figure shows different steps involved in the investigation of the speech recognition application with a recurrent neural reservoir. The input speech samples are pre-processed and noise is removed by an end-point detection technique. Features are extracted and processed as inputs for the neural reservoir. In order to reduce the computational burden of readout, only partial information is extracted and simple feedforward neural network is used for classification.

In the following sections, the approach is investigated and analysed in three stages: feature extraction through linear predictive coding, investigation of stable neural reservoir dynamics and back end processing through simple gradient based learning. The experiment is based on the subset of isolated digit recognition (digits 0-9) dataset from the TI46 speech corpus. The dataset provides samples spoken by five different speakers with each digit being uttered four times by each speaker. This provides a total dataset of 200 speech samples (10 digits x 5 speakers x 4 utterances).

3.1 Pre-processing

In the application of speech recognition, it is very important to detect the signal in the presence of background noise in order to improve the accuracy of a system. A speech signal can be divided into three states: silence, unvoiced and voice (Luh et al., 2004). It is very important to remove the silence state in order to save the overall processing time and hence to improve the accuracy. In order to detect the silence part, an end-point detection technique is used where signal energy is calculated and a threshold value is determined. The total amount of data processing is minimised by accurately detecting the start and stop points in a sample speech signal (see Fig. 7). As shown in the figure, a spoken word 'five' is sampled at 12 KHz for 8260 samples or a duration of 0.69 seconds. Total silence time before and after voice is around 0.37 seconds or 4453 samples. By reducing this silence time, the overall signal pre processing time can be improved to 53%.

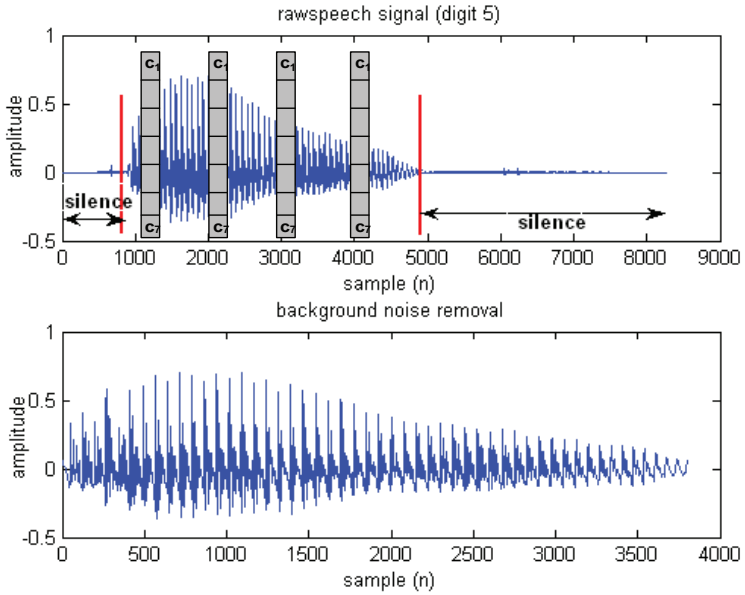


Fig. 7. This figure shows a raw speech signal sampled at 12 KHz for 8260 samples. The utterance can be divided in three clearly differentiable parts: silence, speech and then silence. The waveform has quite a significant part of silence in the beginning and the end of a signal. End-point detection technique removes the silence part from raw speech signal and only voiced portion of the signal as shown in the bottom plot is processed. By using an end point detection technique, overall processing time can be improved to 53%.

3.2 Feature extraction

Once the noise is filtered out from the speech signal, an appropriate speech coding technique is applied for feature selection. Different biologically plausible and signal processing based techniques such as frequential based MFCC (Mel Frequency Cepstral Coefficient), Lyon Passive Ear and Inner Hair Cell models have been reported in the literature and a detailed comparison is provided by (Verstraeten et al., 2005). These

techniques provide a good analysis but none of them offers an optimal solution. In this experiment, a temporal based LPC (Linear Predictive Coding) technique is applied which is one of the most useful methods for encoding a speech signal. In this method, present samples of the speech are predicted by the past *p* speech samples. Mathematically, this can be written as:

$$\tilde{x}(n) = a_1 x(n-1) + a_2 x(n-2) + ... + a_p x(n-p) \tag{3}$$

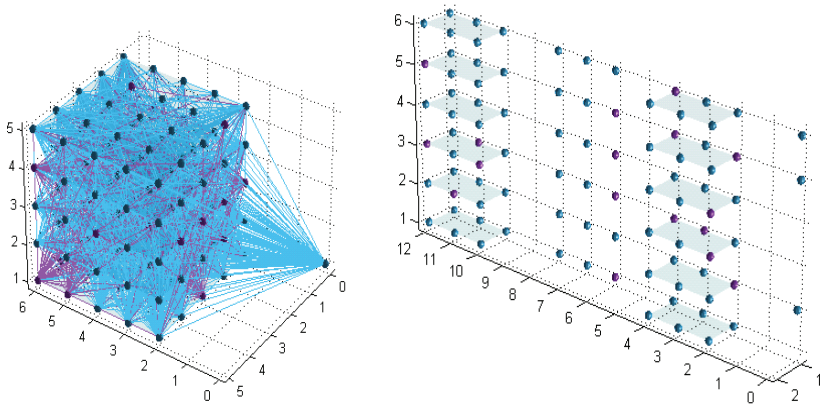
Where $\tilde{x}(n)$ is the predicted signal value, *x(n- p)* the previous observed value and a_p the predictor coefficient. The coefficients, a_1, ….a_p remain constant while the objective is to estimate the next sample by linearly combining the most recent samples. Another important consideration is to minimise the mean square error between the actual sample and the estimated one. The error generated by this estimate can be calculated as:

$$e(n) = x(n) - \tilde{x}(n) \tag{4}$$

Where *e(n)* is the calculated error and *x(n)* is the true signal value.
Speech is sampled at the rate of 12 KHz and noise is removed by the end-point detection technique. The frame size is chosen as 30 *ms* and a frame rate 20 *ms*. Autocorrelation coefficients were computed from the windowed frame and Hamming window is used to minimise the signal discontinuities at the beginning and end of the frame. An efficient Levinson-Durbin's algorithm is used to estimate the coefficients from a given speech signal. It is computationally expensive and not feasible to process all frames in the signal. It also leads to few problems because due to the various signal lengths the total numbers of frames are different. For this experiment, total four frames were selected for each spoken digit in linear distance from the start and end point of the signal, 7 coefficients per time frame over four frames and hence total 28 features per sample were processed. These feature vectors were found to be a good compromise between computational complexity and robustness (Shiraki & Honda, 1988). These frames were used for training and testing the baseline feed forward classifier. For reservoir based approach, same coefficients were used as inputs, further details are reported in the following section.

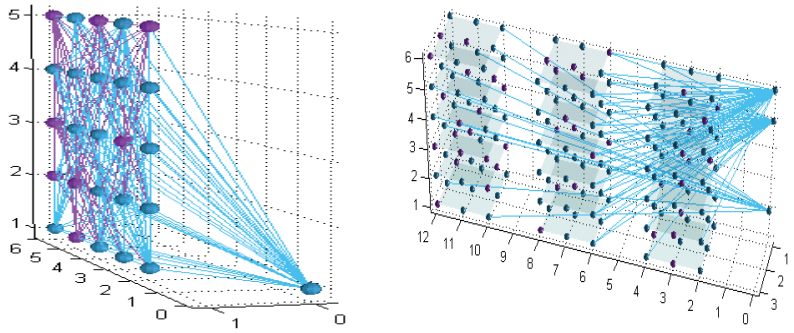
3.3 Reservoir dynamics

In order to model a stable reservoir, it is very important that it should have two qualities: separation and approximation (Maass et al., 2002). The approximation property refers to the capability of a readout function to classify the state vectors sampled from the reservoir. The separation property refers to the ability to separate two different input sequences from each other. This is important, because the readout network needs to be able to distinguish between two different input patterns to have a good classification accuracy. If the output responses of a reservoir for two different inputs are the same then the readout network will not be able to differentiate between the two patterns and thus will not be able to classify which pattern belongs to a particular class.
In this study, reservoirs are generated in a stochastic manner where a 3D column is constructed (see Fig. 8a) which is a biologically plausible way to imitate microcolumnar structures in a neocortex (Maass et al., 2002), other configurations are also possible as shown in Fig. 8 b, 9a and 9b. There have been other strategies proposed in the literature for their satisfactory performance based on empirical data (Legenstein & Maass, 2005) (Skowronski &



(a) (b)

Fig. 8. (a) A 5x5x5 3D grid constructed with a single input neuron
(b) Three microcolumns of size (3x2x6) (3x1x6) and (3x2x6) with three input neurons.



(a) (b)

Fig. 9. (a) A 5x1x5 cortical column fully connected with a single input neuron
(b) A 3x3x6 grid with three input neurons

Harris, 2007) (Jaeger, 2002) (Verstraeten et al., 2007) (Uysal et al., 2007). The design space for constructing a stable reservoir is huge and depends on various important factors such as node type, probability of local and global connections and the size of the reservoir. Apart from the reservoir intrinsic dynamics, there are other factors which contribute to the overall performance of a reservoir such as input features which are used to perturb the reservoir. It is very important that a reliable front end is investigated so that the reservoir can effectively separate different input streams. Once a stable reservoir is constructed then different snapshots are recorded from the reservoir at different time steps and processed for approximating a readout function (Legenstein & Maass, 2005). If two reservoir states $x_u(t)=(R_u)(t)$ and $x_v(t)=(R_v)(t)$ for two different histories $x(.)$ and $v(.)$ are different then the reservoir dynamics are stable, otherwise they will be considered as chaotic. This property is

desirable from practical point of view because different input signals separated by the reservoir can more easily be classified by the readout neurons.
Additionally, the key factor behind a stable reservoir is its short term memory capability which depends on a number of parameters such as membrane threshold, reset voltage and leaky integration. The advantage of bigger reservoirs is that they increase the dimensionality of the reservoir states and data becomes more visible to the readout neurons. One of the criteria in evaluating the computational capability of a recurrent reservoir is to analyse its separation property. It was observed from experimentation that a task which is solved by a large reservoir can also be solved by a much smaller recurrent reservoir provided that the network is capable of differentiating between two different input streams. It is important to observe the reservoir states $x_u(t) = (R_u)\ (t)$ and $x_v(t) = (R_v)(t)$ for two different input histories $u(t)$ and $v(t)$. In order to fulfill the separation property, two different histories have to be captured by the reservoir to prove that the reservoir dynamics are not chaotic. This is also important for a readout function to distinguish between two different inputs in order to approximate the output function. It is demonstrated in this experiment that simple readout neurons such as an MLP could classify different input streams if properly separated by the reservoir. A major advantage of this approach is the improved classification accuracy with a few neurons which overcomes the burden of bigger reservoirs.
The reservoir is constructed as a three dimensional grid (see Fig. 8a) and the probability of connecting two neurons with each other is determined by calculating the Euclidean distance D between the nodes i and j:

$$i = (i_x, i_y, i_z) \text{ and } j = (j_x, j_y, j_z) \tag{5}$$

Where distance between nodes i and j is calculated as:

$$D(i,j) = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2 + (i_z - j_z)^2} \tag{6}$$

In equation 5, i and j are input neurons with three coordinates, x, y and z. The probability to connect two neurons i and j is calculated by using the following equation:

$$P_{conn(i,j)=} C.e\frac{-D(i,j)}{\lambda} \tag{7}$$

In equation 7, λ is used to control the average amount of connections between neurons. Depending on whether neurons i and j were excitatory or inhibitory, the value of C was used as suggested by (Maass et al., 2002). This is an important factor in controlling the reservoir dynamics. Different reservoirs were investigated in order to analyse their short term memory and results are reported. In Fig. 10, an architecture of a cortical column is shown where an input stimulus is fully connected to the reservoir of 27 spiking neurons and states were recorded. The reservoir states $x(t)$ refer to the states of all the neurons in the reservoir in terms of membrane voltages and spike firing times at particular time steps (t, $t+1 \ldots t+n$), further details regarding reservoir states are provided in section 5.
As shown in Fig. 11, most of the neurons in the reservoir are active which shows an ordered activity in response to the input stimulus. These reservoir states can be used as short-term memory for readout neurons. The membrane time constant is set to 30 ms, absolute refractory period to 3 ms and threshold voltage to 15 mV. The reservoir neurons i, j and

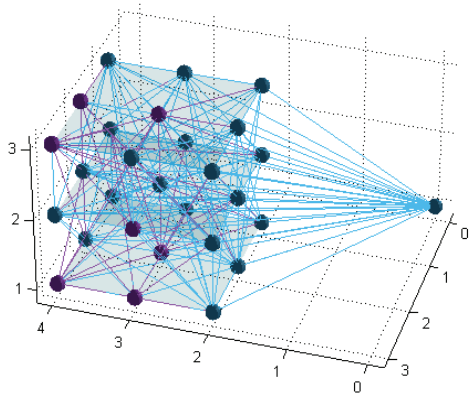


Fig. 10. This figure shows a reservoir of 27 (3x3x3) LIF neurons fully connected with an input neuron. Some neurons are marked with magenta balls which denote inhibitory neurons while others are excitatory neurons.

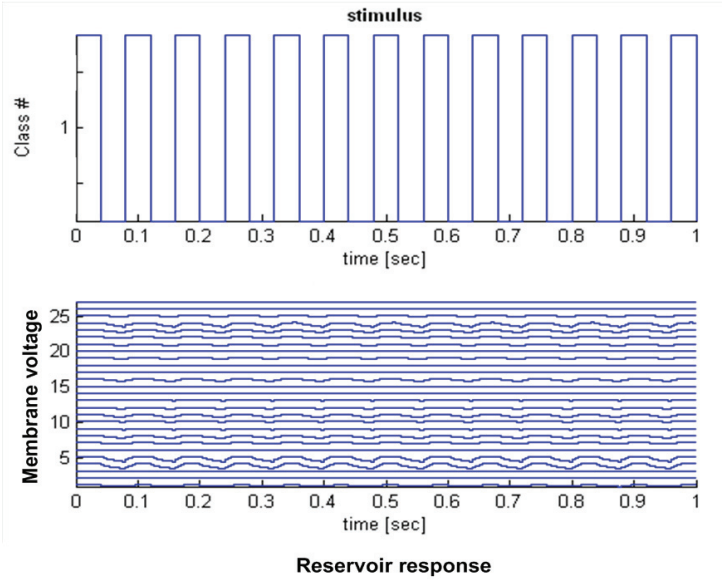


Fig. 11. This figure shows a response of an input square wave in terms of membrane voltages. The top plot shows an input stimulus and bottom plot shows the membrane voltages. The vertical axes show the neurons' membrane potential and the horizontal axis shows the simulation time.

their synaptic connectivity is defined by equation 7 where average amount of connections were controlled by parameter λ. The λ value of 2 is used in these simulations. The parameters selection is based on the biological data obtained from Henry Markram's Lab in Lausanne (Gupta et al., 2003). The data is obtained through experiments on rat somatosensory cortex and suggested by Maass (Maass et al., 2002).

In the paradigm of reservoir computing, readout neurons have an exclusive access to the liquid or reservoir states $x(t)$. For stable reservoir dynamics, it is required that two different inputs $u(s)$ and $v(s)$ should produce two significantly different states $x_u(t)$ and $x_v(t)$ which will hold information about preceding inputs. If the reservoir dynamics are stable (ordered) then a simple memory-less readout can produce the desired output (Natschlaeger et al., 2002). In order to analyse the separation property of reservoirs, various reservoir architectures were simulated and their responses were observed in terms of membrane voltages and spike times. In these simulations, 3D columns were used with different reservoir sizes. A standard leaky integrate and fire model was simulated where the membrane potential of a neuron was calculated as follows:

$$\tau_m \frac{dV_m}{dt} = -(V_m - V_{resting}) + R_m.(I_{syn}(t) + I_{noise}) \tag{8}$$

Where τ_m is the membrane time constant, V_m the membrane voltage, $V_{resting}$ is the membrane resting potential which is 0 V, $I_{syn}(t)$ is the synaptic input current, I_{noise} is a Gaussian random noise with zero mean and a given variance. The membrane potential is set to the value of V_{init} (0.013 V). If the membrane voltage V_m exceeds a certain threshold V_{th} (0.015 V) it is reset to the V_{reset} which is similar to the value of V_{init}.

In order to observe more realistic responses of the reservoir, the features extracted through LPC technique were fed into the reservoir and the reservoir states were recorded. The inputs can be fed into the reservoir in two different ways: analog currents and spike trains. In these simulations analog currents were used as inputs. In section 7, the input values were encoded in spike trains and results were analysed. In Fig. 12, the reservoir architecture is shown with a column of 8 neurons fully connected with an input neuron. The membrane voltages and spike times in response to input digits 1 and 7 in Fig. 13 and 15. For these simulations, the membrane threshold voltages were set to 15 mV and reset voltages to 13.5 mV. An ordered activity is observed in all these responses. It can be seen from these simulations that most of the neuron's membrane voltages and spike firing times were in order which shows a stable reservoir dynamics.

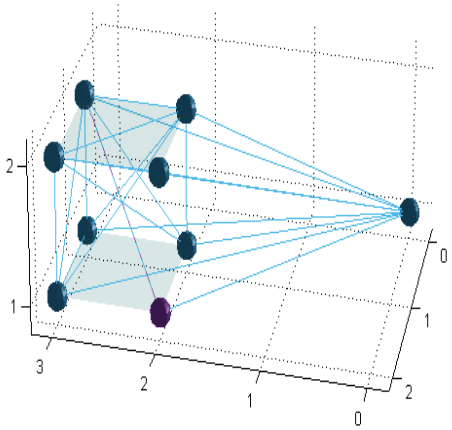


Fig. 12. This figure shows a column of 8 neurons (2x2x2) fully connected with an input neuron. The input to this reservoir is digit 0.

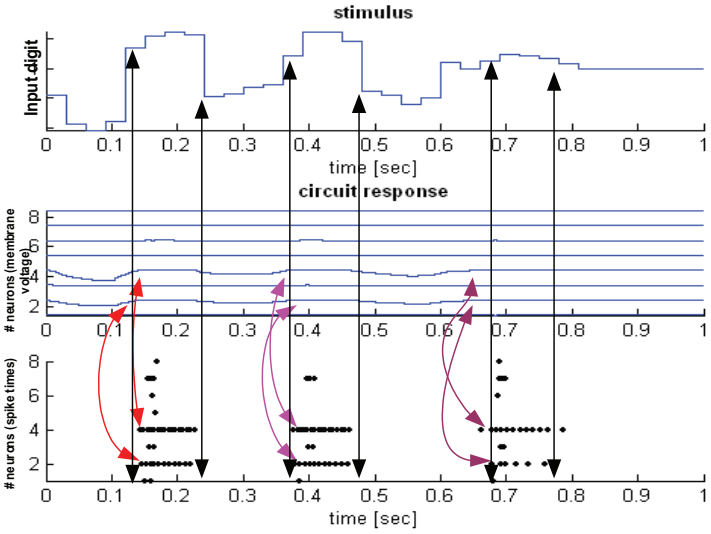


Fig. 13. This figure shows an input digit one and its response in terms of membrane voltage and spike times. The simulation is run for 1 second and spikes were recorded when membrane potential exceeded a threshold value of 15 mV. The reset voltage was set to 13.5 mV. The weights were randomly drawn between -0.1 and 0.1. Total 200 states were recorded by setting the recorder's time step of 5 ms. The middle plot shows the membrane potential and bottom plot shows the spike times. The membrane activity is shown in comparison with the spike firing times.

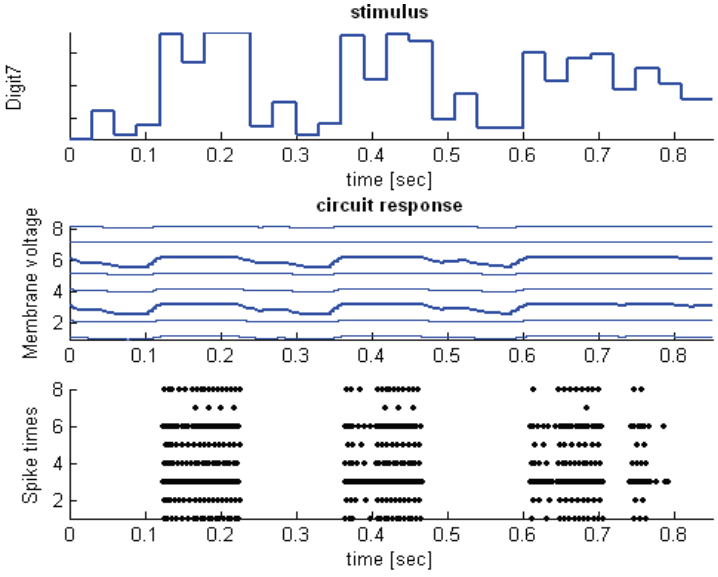


Fig. 14. This figure shows the internal dynamics of a reservoir in response to digit 7.

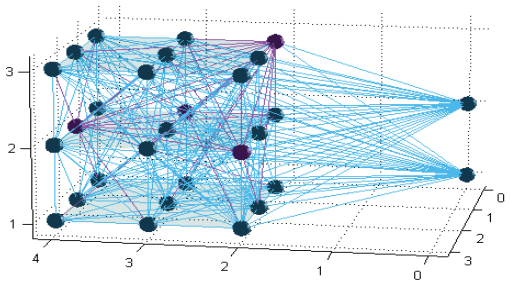


Fig. 15. This figure shows an architecture of 27 (3x3x3) recurrent LIF neurons fully connected with two input neurons.

In these simulations the reservoir architecture and the number of neurons were remained fixed and internal states were analysed in response to different input stimuli. It is important to observe that reservoir internal states correspond to the input stimulus and could separate two different inputs. This is an important property which has to be verified for successful classification because readout neurons will have an exclusive access to the membrane voltages of the neurons in the reservoir, if the reservoir states were not significantly different from each other then the readout neurons will not be able to classify different inputs.
In order to analyse the robustness of a reservoir, two inputs were simultaneously applied to the fully connected reservoir and states were recorded. The state differences were analysed as shown in Fig. 18 and 19. Fig. 17 shows the architecture used for these simulations with 27 LIF neurons fully connected with two input neurons. The inputs to the reservoir are two sine waves with different frequencies.

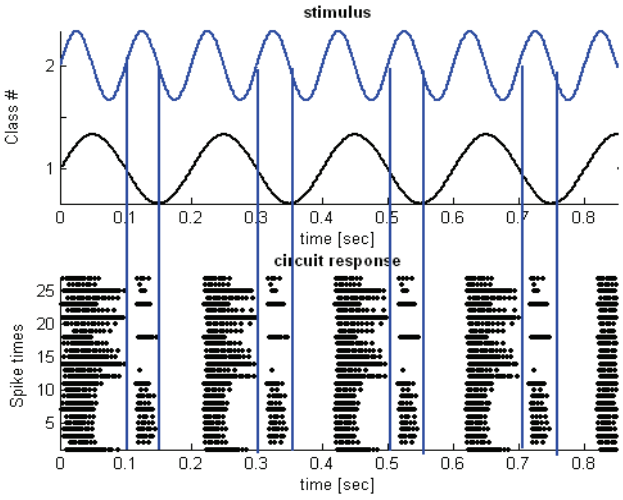


Fig. 16. This figure shows spike times in the bottom plot in response to two different inputs (top plot). The vertical blue lines show the neurons firing times in response to input 2 which is clearly separable from input 1.

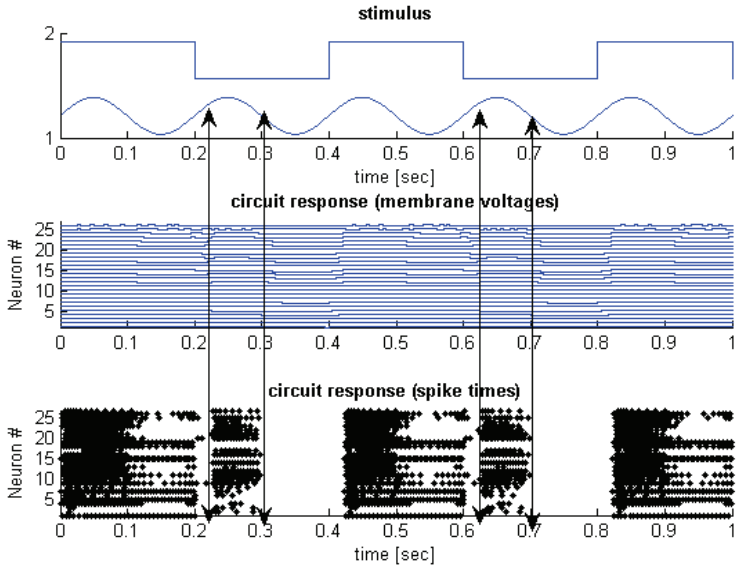


Fig. 17. This figure shows the membrane voltages and spike times in response to two different inputs. The reservoir activity is in order with regard to the input stimuli where both spike times and membrane voltages are separable.

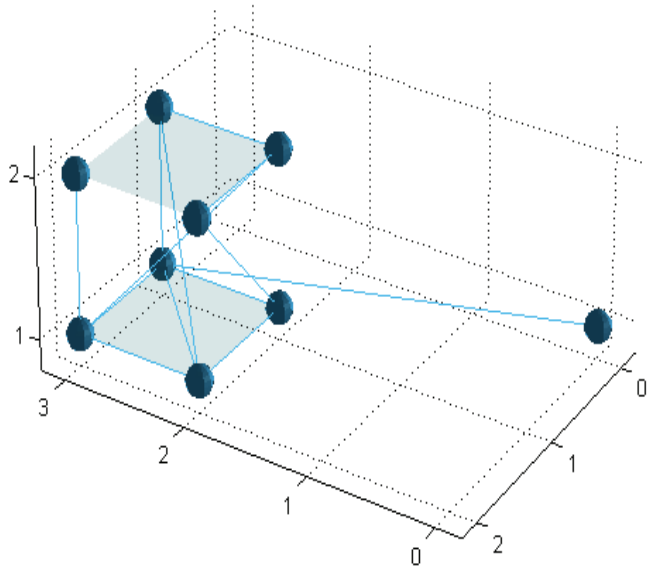


Fig. 18. This figure shows an architecture of 8 neurons (2x2x2) partially connected with an input neuron.

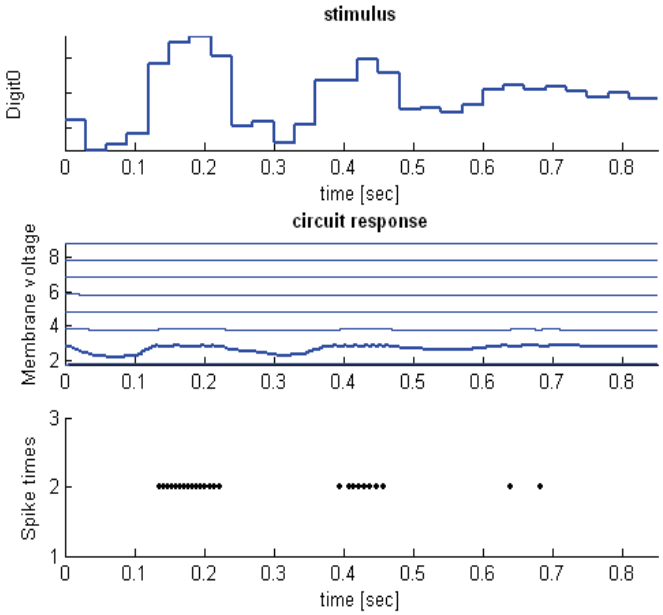


Fig. 19. This figure shows a reservoir response in terms of membrane voltages and spike times. Due to the limited connectivity, only one neuron triggered to fire.

Information processing in cortical neurons crucially depends on their local circuit connectivity. Many efforts have been made to investigate the neuronal wiring of cortical neurons and is still an active area of research (*Braitenberg & Schuz, 1998*) (*Holmgren et al., 2003) (Gupta et al., 2000) (Foldy et al., 2005) (Yoshimura et al., 2005)*. The overall state of a reservoir very much depends on the connectivity of input neurons with the reservoir. For partially connected inputs, it is less likely that most of the neurons in the reservoir will have short term memory, however, for fully connected inputs, the probability of neurons having the short term memory increases many folds. A series of experiments carried out with different input connectivity and results are reported. In Fig. 18, an architecture is shown with a reservoir size of 8 neurons where input is connected with the reservoir with just one connection. The stimulus to the reservoir is digit 0 and states were recorded in terms of membrane voltages and spike times. An extremely low activity is observed as shown in Fig. 19, however, the short term memory of a reservoir increases with increased input connectivity as shown in Figs. 21 and 23. The reservoir architectures are shown in Figs. 20 and 22.

Input connectivity is an important design decision and affects the overall accuracy of the classifier because reservoir states will be used as training vectors for readout neurons. If the reservoir states were not properly recorded then regardless of the size of the reservoir, the readouts will not be able to classify.

This section investigated different reservoir topologies and dynamics with the help of routines from the CSIM toolbox (Natschlaeger et al., 2002). In section 4, state vectors recorded from the reservoir will be used for training the neural classifier (backend) in order to check the accuracy and robustness of the classifier. The rationale behind this investigation was to analyse the reservoir dynamics which is crucial for readout neurons. It is far from

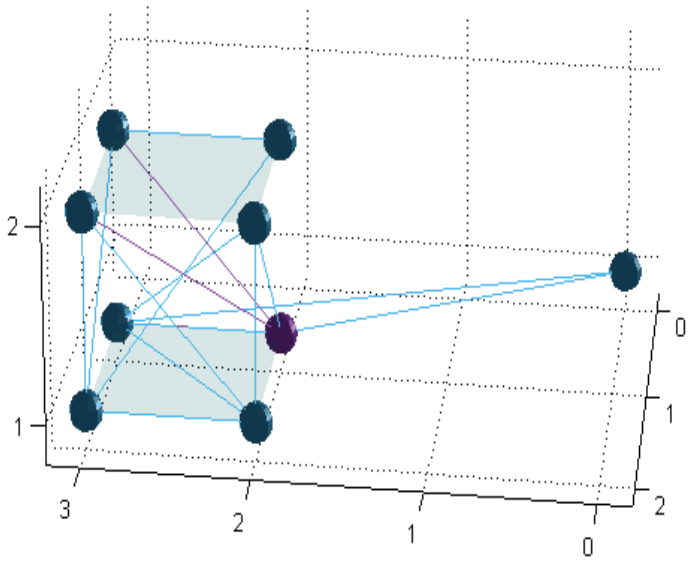


Fig. 20. This figure shows an improved connectivity where input neuron is connected with reservoir with two connections.

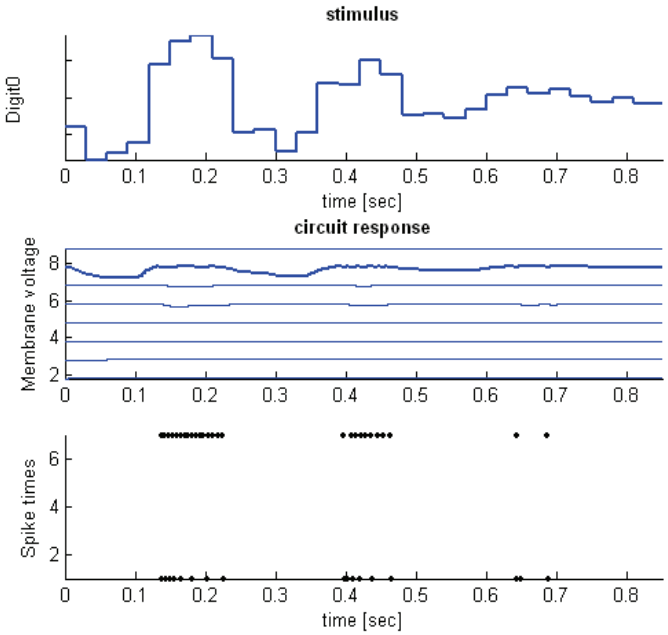


Fig. 21. This figure shows a response of reservoir with an improved neuron activity.

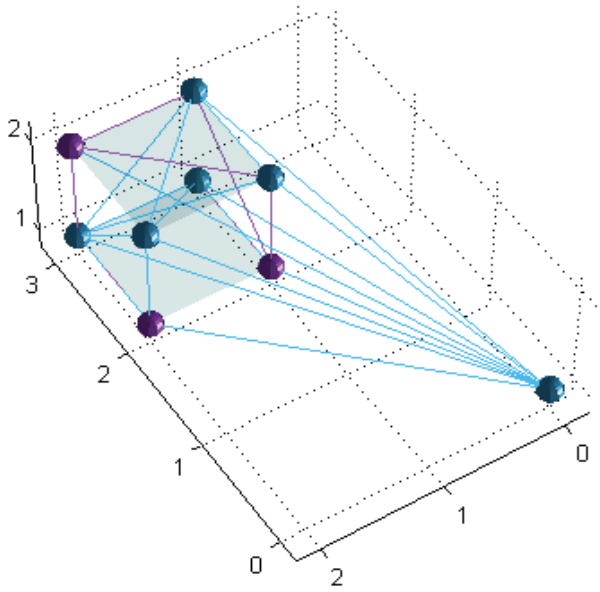


Fig. 22. This figure shows a fully connected network where input neuron is fully connected with all the neurons in the reservoir.

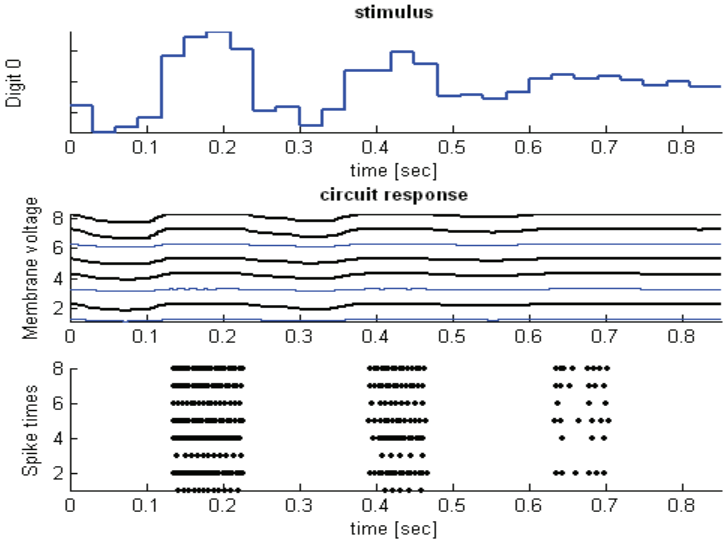


Fig. 23. This figure shows that almost all the neurons are active in the reservoir and the reservoir dynamics are ordered and correspond to the input stimulus.

trivial to model a stable reservoir and very much depends on the experience with this paradigm. The parameters selected for these simulations were empirical, however, once a stable reservoir is modelled, it can successfully be used for different input stimuli and the memory capacity (maximum number of patterns that could be stored for a short period of time) can easily be improved by increasing the size of the reservoir without any significant degradation. In the following section, a baseline feed forward MLP classifier is implemented and features extracted from speech front end are used for training and testing.

The neocortex constitutes almost 80% of the brain and appears to be the fundamental unit of information processing in mammals. The neural microcircuit plays an important role in functions such as adaptability, memory and higher cognitive functions (Natschlager et al., 2003). This study emphasises the importance of connectivity and compact stable reservoirs for microcircuit design. The most interesting characteristic of the neocortical microcircuit is their outstanding computational power where temporal aspect is not necessary for the training of readout neurons because temporal processing is done only at the reservoir level. The biggest advantage of neural microcircuit is that it doesn't require any task specific connectivity, rather the same circuit can be used for different computational tasks and only readouts are required to be trained to produce desired outputs. It is theoretically analysed and predicted by Mass (Maass et al., 2002) that there are no limitations to the power of this model and it can be used as a universal computational model, however, one needs to investigate a reservoir which will work as a fading memory and fulfill the separation property.

4. MLP baseline classifier

The paradigm of reservoir computing very much depends on suitable techniques for front and back end processing. In the previous experiments, front end and reservoir dynamics were investigated; this section investigates a feed forward MLP classifier for backend processing. In order to evaluate classification accuracy, different network sizes and standard backpropagation learning algorithms (backpropagation, resilient backpropagation and Levenberg-Marquardt) were evaluated.

In order to pre-process input data, four frames were selected to extract coefficients from an input speech signal with 7 coefficients per frame and 28 features per sample. The features were extracted through an LPC technique as stated earlier. The pre processing is done in order to remove the noise from a signal and then coefficients were extracted at the frame rate of 20 ms and analysis is done by windowing the speech data with a window size of 30 ms. The noise removal is performed with a threshold technique where threshold is compared with the standard deviation of the signal power.

For this experiment, the total dataset consisted of 200 samples, divided into two sets (training and testing), 20 samples (5 speakers x 4 utterances) for each digit with 28 LPC features per sample. In order to analyse the classification accuracy, different training sets and hidden layer neurons were investigated. The results in terms of classification accuracy are shown in Tabs. 1 and 2. In a series of experiments, the best results obtained in those trials are shown in the following tables. For performance evaluation, the Matlab gradient descent with adaptation (traingda) training algorithm was used where learning rate was adjusted to the value of 0.1 and the goal was set to 0.01.

It is observed from these experiments that by increasing the number of training samples and hidden neurons the overall accuracy improves except the decrease in performance when numbers of hidden neurons were increased from 20 to 25 in Table 1 and from 25 to 30 in

Hidden neurons	Digit 1 (%)	Digit 2 (%)	Digit 3 (%)	Digit 4 (%)	Digit 5 (%)	Digit 6 (%)	Digit 7 (%)	Digit 8 (%)	Digit 9 (%)	Digit 0 (%)	Mean (%)
10	100	60	60	60	40	40	20	0	40	80	50
20	100	40	80	60	80	60	20	20	60	100	62
25	80	60	60	60	80	40	20	20	60	60	54
30	100	60	60	60	80	60	40	20	40	100	62
35	80	60	80	80	100	80	0	40	80	80	68

Table 1. Total samples = 150, training samples = 100, test samples = 50

Hidden neurons	Digit 1 (%)	Digit 2 (%)	Digit 3 (%)	Digit 4 (%)	Digit 5 (%)	Digit 6 (%)	Digit 7 (%)	Digit 8 (%)	Digit 9 (%)	Digit 0 (%)	Mean (%)
6	80	20	100	100	60	60	20	20	80	100	58
20	80	40	100	40	60	80	20	20	80	100	62
25	100	40	100	80	80	80	40	20	80	100	72
30	100	40	100	80	80	80	20	20	80	80	68
30	100	40	100	80	100	80	20	60	80	100	76
35	100	40	100	80	80	80	40	20	80	100	72

Table 2. Total samples = 200, training samples = 150, test samples = 50

Table 2, however the maximum performance obtained was limited to 76%. By increasing the hidden neurons more than 35 the overall performance starts decreasing. The investigation of feed forward network was motivated due to two reasons: first to evaluate the performance of feed forward networks as a standalone classifier, and second to observe the bottlenecks because the same classifier will be used as readout for the reservoir where features extracted from the reservoir will be used as training sets.

5. Reservoir based recognition

The previous sections investigated the front end (speech pre processing), backend (feed forward network) and neural reservoir dynamics. This section will investigate the realization of the complete paradigm of reservoir computing by integrating the overall components investigated in previous sections (see Fig. 24). The paradigm operates by feeding input features extracted through LPC technique into the recurrent neural reservoir as the post synaptic currents. The reservoir is constructed from a random recurrent spiking neural network which projects linearly non separable low dimensional inputs to a high dimensional space. The recurrent neural network by itself is not trained, rather, different reservoir states sampled every 25 ms were recorded for backend classification by the MLP (see Fig. 25). Theoretically, there is no limit to the computing power of this paradigm if the reservoir dynamics are stable and fulfill the requirements of short term memory and state separation. If the reservoir dynamics are carefully analysed then a simple gradient based algorithm can successfully classify a complex problem such as speech recognition. In this experiment, standard supervised algorithms were investigated to quantify their classification accuracies based on the input features sampled through the reservoir. In order to analyse the separation property, different reservoir sizes were chosen and classification accuracies were calculated. The LIF neurons were used for reservoir construction. Both static and dynamic

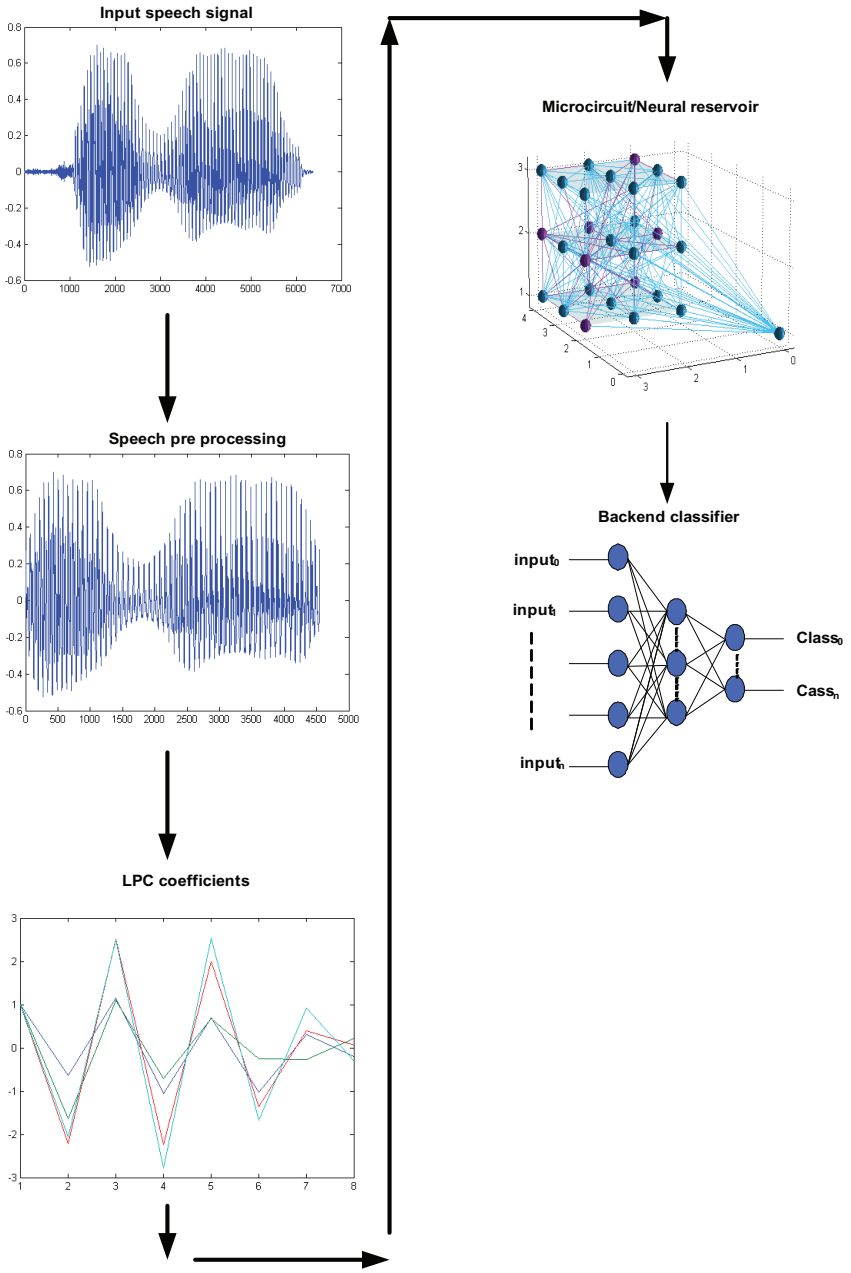


Fig. 24. This figure shows a proposed reservoir based speech recognition approach where input speech signals are pre processed and data was fed into the recurrent neural reservoir. The learning is performed by the feed forward readout neurons.

synapses were used amongst which 20% were chosen to be inhibitory as suggested by (Maass et al., 2002) based on the data from Henry Markram's Lab in Lausanne (Markram, Lausanne). Once the reservoir is perturbed with the input stimuli, the state vectors were recorded from the reservoir sampled at 25 ms and used for training the feed forward MLP classifier with a single hidden layer and 10 output neurons. The recorder time was set to 5 ms and for 1 second simulation of a reservoir total 200 states were recorded. This state vector contained the membrane voltages of all the neurons in the reservoir at each recording time step. It is neither required nor feasible to process all these states for backend classification, therefore these states were sampled at 25 ms in linear scale from start to the end of simulations and used as training vectors for the classifier. In all these experiments, only the readout neurons were trained whereas the reservoir connectivity remained fixed for generating the reservoir states. The performance of backend feedforward classifier was evaluated with test samples and the best results obtained in different trials are shown in Table 3, 4 and 5.

In Table 3, a reservoir size of 8 neurons successfully classified the input data and the results shown in Tables 4 and 5 also correspond to the theoretical framework of reservoir computing. It is evident from these experiments that for better classification accuracies, stable reservoirs are more important than merely the size of the reservoir. These results also justify the Cover's separability theorem (Cover, 1965). Using backpropagation algorithms

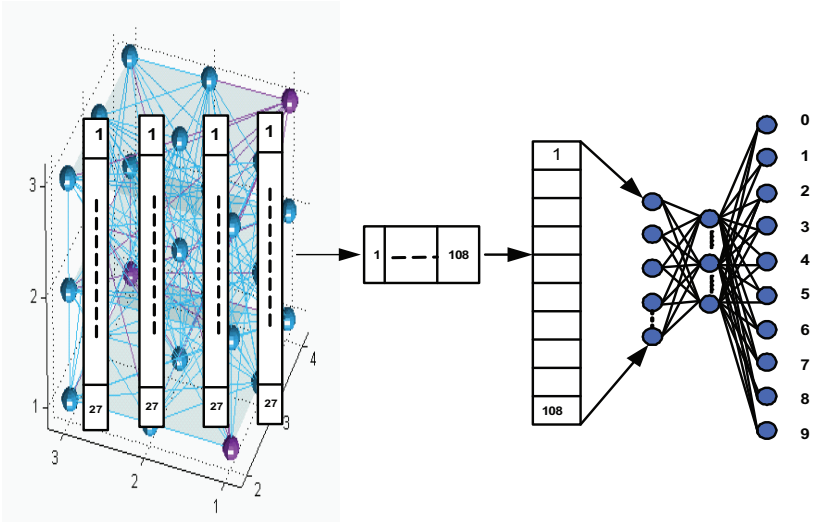


Fig. 25. Processing all frames in the reservoir are computationally expensive therefore specific frames are selected in linear distance with reference to the start point and the end point of the simulations. Each frame consists of the total number of neurons in the reservoir sampled at the rate of 25 ms.

Readout	Reservoir size	Network structure	Test accuracy (%)
Matlab RP	8	32-20-10	94.8
Matlab LM	8	32-30-10	100
Matlab BP	8	32-50-10	96

Table 3. Test performance with reservoir size = 8

Readout	Reservoir size	Network structure	Test accuracy (%)
Matlab RP	15	60-20-10	80.4
Matlab LM	15	60-30-10	98.8
Matlab BP	15	60-50-10	92.8

Table 4. Test performance with reservoir size = 15

Readout	Reservoir size	Network structure	Test accuracy (%)
Matlab RP	27	108-20-10	100
Matlab LM	27	108-30-10	96
Matlab BP	27	108-50-10	100

Table 5. Test performance with reservoir size = 27

for readout neurons are advantageous because they can approximate complicated target functions if appropriate network architectures were selected. In this study, different architectures were investigated with different hidden layers and number of neurons and results are reported. A thorough discussion about these algorithms is out of the scope of this chapter, reader is refer to (Hertz et al., 1991) (Duda et al., 2001) for further details. There is no specific learning algorithm which will guarantee a good accuracy of the readout neurons. A logical approach is to start with standard backpropagation algorithms and test the accuracy with different hidden layers and number of neurons. The good classification accuracy will not only infer to the suitable architecture of a feed forward neural network but will also show that the reservoir has significantly separated different inputs and have projected data sufficiently on a high dimensional space. These series of experiments have thoroughly investigated the theoretical framework of reservoir computing and results are demonstrated with an speech recognition application. In order to quantify results, a baseline feedforward classifier is implemented and results are compared with the reservoir based technique. This section has investigated the framework with analog inputs extracted through speech front end, in the following section, an experiment is carried out where input stimuli is converted into Poisson spike trains and results were analysed.

6. Spike based coding

Biologically plausible neurons communicate through spike trains, this section will investigate the framework by encoding the analog input values into spike trains and results are reported. As stated by Squire and Kosslyn that the timing of successive action potentials is irregular in the cortex (Squire & Kosslyn, 1998), therefore Poisson spike coding technique is used where the generation of each spike was dependent only on an underlying analog driving signal and each spike was considered to be independent of all the other spikes (Heeger, 2000).
In order to generate a spike train, an interspike interval is randomly drawn from an exponential distribution and each successive spiketime is calculated by the previous spiketime plus a randomly drawn interspike interval. In order to convert input features into Poisson spike trains, first negative analog values are converted into positive values and spike times were calculated (see Fig. 26). The spike times were sampled at 100 ms for maximum time in order to generate spike trains (see Fig. 27).

The spiketimes calculated through Poisson encoding were fed into the reservoir and states were recorded. The total numbers of recorded states were kept the same for fair comparison with analog inputs as stated in section 5. The results are shown in Table 6, 7 and 8. The reservoir is tested with the size of 8 and 15 neurons and the overall accuracy didn't improve by increasing the size of reservoir. The best accuracy achieved with Poisson encoding was limited to 98%.

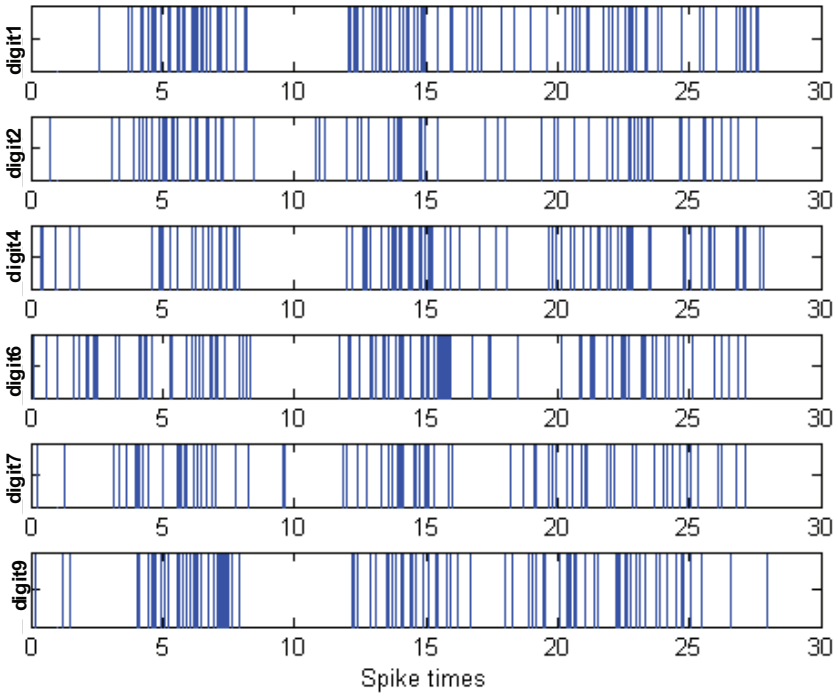


Fig. 26. Spike times with Poisson encoding for digit1, 2, 4, 6, 7, and 9

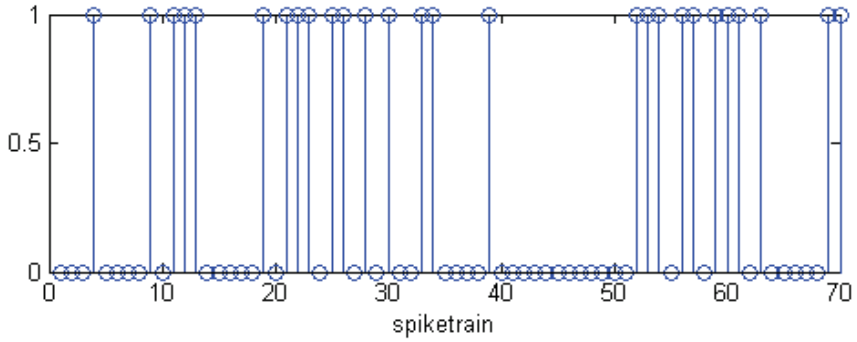


Fig. 27. Spike times sampled at rate 0.1 for maximum time in order to generate spike trains

Readout	Reservoir size	Network structure	Test accuracy (%)
Matlab RP	8	32-20-10	62
Matlab LM	8	32-30-10	98
Matlab BP	8	32-50-10	80.4

Table 6. Test performance with reservoir size 8

Readout	Reservoir size	Network structure	Test accuracy (%)
Matlab RP	15	60-20-10	70.3
Matlab LM	15	60-30-10	78
Matlab BP	15	60-50-10	77.5

Table 7. Test performance with reservoir size 15

Readout	Reservoir size	Network structure	Test accuracy (%)
Matlab RP	27	108-20-10	72
Matlab LM	27	108-30-10	71.5
Matlab BP	27	108-50-10	75.3

Table 8. Test performance with reservoir size 15

In this experiment, Poisson spike trains were used as input for reservoir but no significant improvement is achieved by increasing the reservoir size and accuracy found to be inferior than previous experiments where analog values were used. The possible reason is due to the highly random Poisson process, the spike trains were randomly generated and reservoir couldn't differentiate between different spike trains. The results vary from one trial to another and best results achieved from those trials are reported in Table 6, 7 and 8.

7. Summary

One of the key properties of the reservoir computing is their short term memory and ability to separate different inputs. This property is called fading or echo state property which is very much dependent on the internal dynamics of the reservoir. The memory capacity of bigger reservoirs can only be useful if reservoir dynamics are not chaotic. If the reservoirs are not stable then regardless of the size, memory characteristics can not be achieved. The minicolumns in the cortex appear to be the basic unit of computing where each microcolumn consists of few neurons (Mountcastle, 1997). The columns are distributed therefore these experiments suggest that small reservoirs with stable dynamics are more reliable than bigger chaotic reservoirs. Once an optimal reservoir is investigated, backend processing can further improve the overall performance.

The rationale behind the experiments conducted in this work was not to exhaustively check the readouts or suitable front ends, rather to investigate the theoretical framework and its viability as a universal classifier. Recently, different aspects of the reservoir computing such as mean field theory, edge of chaos, biologically plausible front ends, computational nodes and memory capacity have been investigated and reported in (Maass et al., 2002) (Skrownski et al., 2007) (Jaeger, 2001) (Verstraeten et al., 2007) and (Uysal et al., 2007). The

hybrid implementation presented in this chapter is more suitable for the framework of reservoir computing and the results provided support the theoretical framework.
Given the complexity of the speech recognition problem, the paradigm was split into three sub sections, front end, back end and optimal reservoir. The components were implemented and analyzed individually and integrated for several final experiments. Most of the parameter selection in these experiments were empirical and depends on the experience related to the reservoir computing. It is obvious from these experiments that pre and post processing are important factors because reservoir computing can not guarantee to perform well if either the front or backend are not properly selected. Despite the promising results obtained through this investigation, a fundamental question remains open regarding the way data is pre processed in this study and other related work. In SNNs, pre processing may not be the best way to communicate with spiking neurons and this is the fundamental area that needs further investigation and which is outside the scope of this book chapter.
This chapter thoroughly investigated the theoretical framework of reservoir computing and extended by analysing the compact reservoir dynamics, front end pre processing and back end classification technique. The reservoir based recurrent neural architectures have proven to perform better on classification related tasks such as speech recognition, however, their performance can be increased if combined with feed forward networks. An alternative approach is proposed by utilizing the idea of reservoir computing with efficient feature extraction technique and learning by rather simple feed forward network. This framework revealed a powerful alternative for recognition task and provides a significant improvement in terms of their performance and robustness. To the best of authors' knowledge, none of the existing reservoir based techniques successfully classify the speech recognition problem with an extremely compact reservoir. This study emphasized the modelling of a compact dynamic reservoir and empirically investigated the short term memory capacity and separation property for stable reservoirs. The results show that a stable reservoir and efficient front end technique can solve significantly complex recognition task with simple readouts.

8. References

Haykin, S (1999), Neural Networks, A comprehensive foundation, 2 ed, Prentice hall, Inc, New Jersey.

Pavlidis N.G, Tasoulis D.K, Plagianakos V.P, Nikiforidis G and Vrahatis M.N. (2005), Spiking neural network training using evolutionary algorithms, International Joint Conference on Neural Networks, *IJCNN'05,* pp. 1-5.

Bohte S., Kok J and La Poutre H (2000), Spike-prop: Error backpropagation in multi-layer networks of spiking neurons, *Euro Symposium on Artificial Neural Networks ESANN'2000,* pp. 419–425.

Maass, W, Natschläger, T and Markram, H (2002), Real-time computing without stable states: A new framework for neural computation based on perturbations, *Neural Computation, 14(11),* pp. 2531-2560.

Jaeger, H. (2001), The "echo state" approach to analysing and training recurrent neural networks, Tech. Rep, Fraunhofer Institute for Autonomous Intelligent Systems, *German National Research Center for Information Technology,* (GMD Report 148)

Joshi, P and Maass, W (2005), Movement generation with circuits of spiking neurons, *Neural Computation, 17(8),* pp. 1715-1738.

Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science 304(5667)*, pp. 78–80 (2004).

Skowronski, M.D., Harris, J.G (2007).: Automatic speech recognition using a predictive echo state network classifier, *Neural Networks 20(3)*, pp. 414–423.

Verstraeten, D., Schrauwen, B., D'Haene, M., Stroobandt, D. (2007), An experimental unification of reservoir computing methods, *Neural Networks 20*, pp. 391–403.

Uysal, I., Sathyendra, H., Harris, J.G. (2007), Spike based feature extraction for noise robust speech recognition using phase synchrony coding, *International Symposium on Circuits and Systems,* pp. 1529–1532.

Hopfield, J. J. (1982) Neural networks and physical systems with emergent collective computational properties, *Proc. Nat. Acad. Sci. (USA) 79*, pp. 2554-2558.

Elman, J.J (1990), Finding structure in time, *Cognitive Science, 14*, pp. 179 – 211.

Jordan, MI (1996), Neural networks, A Tucker editors, *CRC handbook of computer science, CRC press.*

Bengio, Y (1996), Neural networks for speech and sequence recognition, *Book published by International Thomson Computer Press.*

Looney, C G (1997), Pattern recognition using neural networks, theory and algorithms for engineers and scientists, *book published by OUP USA*.

K. Doya (1995), Recurrent Neural Networks, supervised learning in M.A Arbib Editors, *The Handbook of Brain Theory and Neural Networks, MIT press*.

Atiya, A. F and Parlog, A G (2000), New results on recurrent network training: unifying the algorithms and accelerating convergence, *IEEE Transactions in Neural Networks: 11(3)*, pp. 697-709.

Pearlmutter, B.A (1995) Gradient calculation for dynamic recurrent neural networks: a survey, *IEEE Transactions on Neural Networks: 6(5)*, pp. 1212 -1228.

Jaeger, H. (2002), Adaptive nonlinear system identification with echo state Networks, *Advances in neural information processing systems*, pp. 593–600.

Williams, RJ and Zipser, D (1989), A learning algorithm for continually running fully recurrent neural networks, *Neural Computations, 1, pp.* 270 - 280.

S. Singhal and L. Wu. Training feed-forward networks with the extended Kalman filter, *IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 1187.1190, 1989

Atiya, A. F and Parlog, A G (2000), New results on recurrent network training: unifying the algorithms and accelerating convergence, *IEEE Transactions on Neural Networks: 11(3), pp.* 697-709.

Abbott, L.F and Nelson S.B (2000), Synaptic plasticity, taming the beast, *Nature Neuroscience, 3(supplement),* 1178-1183.

Steil, J.J (2004, Backpropagation-Decorrelation: online recurrent learning with *O(N)* Complexity, *International Joint Conference on Neural Networks, vol. 1*, pp.843–848 (2004).

Maass, W., Natschläger, T and Markram, H (2004), Computational models for generic cortical microcircuits, *Computational Neuroscience: A Comprehensive Approach, Chapter 18*, pages 575-605

Hopfield, JJ and Brody, CD (2000), What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration, *PNAS (98)*, pp. 1282-1287.

Cover, T.M (1965), "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition", *IEEE Transactions on Electronic Computers, EC -14(3),* pp. 326 – 334.

Sivakumar, SC., Phillip, WJ and Robertson, W (2000), Isolated digit recognition using a block diagonal recurrent neural network, *Canadian conference on electrical and computer engineering,* pp. 726 – 729

Zhao, Z (1991), Connectionist training of non-linear hidden Markov models for speech recognition, *IJCNN,* pp. 1647 – 1652.

Kim, SH., Koh, SY., Ahn, JY and Hur, KI (1999), A study on the recognition of the isolated digits using recurrent neural predictive HMMs, TENCON, pp. 593 – 596

Luh, C and Jantan, A (2004), Digit recognition using neural networks, *Malaysian Journal Of Computer Science, Vol. 17*, pp. 40-54.

Verstraeten, D, Schrauwen, B., Stroobandt, D and Campenhout, JV (2005), Isolated word recognition with the liquid state machine: a case study, *Information Processing Letters 95*, pp. 521 – 528.

Shiraki, Y and Honda, M (1988), LPC speech coding based on variable-length segment quantization, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9), pp. 1437 -1444.

Legenstein, R and Maass, W (2005), What Makes a Dynamically System Computationally Powerful? New Directions in Statistical Signal Processing: From Systems to Brain, *MIT press.*

Gupta A, Wang Y, Markram H (2003) Organizing principles for a diversity of GABAergic Inter-neurons and synapses, *Neocortex Science (287)*, pp. 273–278.

Thomas Natschläger, Wolfgang Maass (2002), Spiking neurons and the induction of Finite State Machines, *Theor. Comput. Sci. 287(1)*: pp. 251-265.

Squire, L.R., Kosslyn, S.M. (1998), Findings and current opinion in cognitive neuroscience, *The MIT Press*, USA

Braitenberg, V., Schuz, A. (19989): Anatomy of the Cortex: Statistics and Geometry, *Sprin ger-Verlag*.

Mountcastle, V B(1978), An organizing principle for cerebral function: the unit model and the distributed system, in the mindful brain, *MIT press*, Cambridge, MA.

Doddington, G.R., Schalk, T.B. (1981), Speech recognition: Turning theory to practice, *IEEE Spectrum* 18(9).

Foldy, C., Dyhrfjeld-Johnsen, J., Soltesz, I. (2005), Structure of cortical microcircuit theory, *J. Physiol. 562*, 47–54.

Yoshimura, Y., Dantzker, J.L.M. (2005), Callaway, E.M.: Excitatory cortical neurons form fine-scale functional networks, *Nature 433*, 868–873.

Gupta, A., Wang, Y., Markram, H. (2000), Organizing principles for a diversity of GABAer gic interneuron and synapses, *Neocortex. Science* 287, pp. 273–278.

Holmgren, C., Harkany, T., Svennenfors, B., Zilberter, Y. (2003), Pyramidal cell communicat ion within local networks in layer 2/3 of rat neocortex, *J. Physiol.* 551, pp. 139–153.

Braitenberg V, Schuz A (1998) Cortex: Statics and Geometry of Neuronal Connectivity, book published by *Springer-Verlag*, Berlin.

Ghani, A., McGinnity, T.M., Maguire, L.P., Harkin, J.G. (2006), Analyzing the framework of 'Reservoir Computing' for hardware implementation, *NIPS workshop on Echo State Networks*, pp. 1–2.

Arfan Ghani, T. Martin McGinnity, Liam P. Maguire, Jim Harkin (2008), Neuro-inspired Speech Recognition with Recurrent Spiking Neurons, *ICANN (1)*, pp. 513-522.

3

The Effect of Reverberation on Optimal GMM Order and CMS Performance in Speaker Verification Systems

Noam R. Shabtai, Boaz Rafaely and Yaniv Zigel
Ben-Gurion University of the Negev, Israel

1. Introduction

In speaker recognition, features are extracted from speech signals to form feature vectors, and statistical pattern recognition methods are applied in order to model the distribution of the feature vectors in the feature space. Speakers are recognized by pattern matching of the statistical distribution of their feature vectors with target models. Speaker verification (SVR) is the task of deciding, upon receiving tested feature vectors, whether to accept or reject a speaker hypothesis, according to the speaker's model. A popular feature extraction method for speech signal processing is the *mel-frequency cepstral coefficients* (MFCC) [Davis & Mermelstein, 1980], and *Gaussian mixture models* (GMM) has become a dominant approach for statistical modeling of speech feature vectors for text-independent SVR [Reynolds et al., 2000].

Speaker verification is widely used in telecommunication or conference room applications, where reverberation is often present due to the surrounding room environment. The presence of reverberation adds distortion to the feature vectors, which results in performance degradation of SVR systems due to mismatched conditions between trained models and test segments.

Feature normalization techniques such as the *cepstral mean subtraction* (CMS) [Mammone et al., 1996] and variance normalization [Chen & Bilmes, 2007], and score normalization techniques such as the Znorm, Hnorm, Tnorm [Bimbot et al., 2004, Mammone et al., 1996] and Top-norm [Zigel & Wasserblat, 2006], were originally developed to compensate for the effect of a telephone channel [Mammone et al., 1996], or for the effect of slowly varying convolutive noises in general [Reynolds et al., 2000]. For that reason, these techniques may be used to reduce the effect of reverberation, if it is characterized by a short-duration *room impulse response* (RIR). However, it may be difficult to find research studies in the literature on the effect of CMS on SVR performance under reverberation conditions of long duration RIR, which is often the case in room acoustics.

In cases of long-duration RIR, the target models may be trained using a reverberant speech database, as suggested by Peer et al. [Peer et al., 2008], in order to overcome the mismatched conditions between the models and the reverberant testing speech segments. This method was tested on *adaptive-GMM* (AGMM) based SVR system, with various values of *reverberation time* (RT - the time that takes the impulse response to decay by 60dB [Schroeder,

1965]). Matching of RT between train and test data was reported to reduce the *equal error rate* (EER) from 16.44% to 9.9% on average, when using both Znorm and Tnorm score normalizations.
The methods that were described in the previous paragraph used fixed GMM order, and were automatically performing feature normalization. This chapter shows that the effect of reverberation on the feature vectors might decrease the optimal GMM order, for *Bayesian* and *Kullback information criteria* (BIC and KIC, respectively). As a feasibility study, the relatively simple case of GMM without adaptation was used, as currently AGMM systems are designed for using constant model order. However, the study in this chapter might suit a future adjustment of AGMM systems.
The investigation of the effect of GMM order is based on a study performed by the authors [Shabtai et al., 2008a], where only simulated RIRs were used. This chapter also investigates the effect of reverberation on the performance of CMS applied to MFCC feature vectors in SVR. In that sense, it serves as an extension of an early study of the authors [Shabtai et al., 2008b], where only simulated RIRs were used to form reverberant speech. Here both simulated and measured RIRs are employed.

2. Room parameters

Room parameters can either have a direct relation to the physical characteristics of the room, or some relation to the RIR. Associated with the physical characteristics of the room we have the geometrical characteristics, which are the volume V and the surface area S, and the reflection coefficient of the room boundaries, R. The absorption coefficient of the room boundaries a is defined as [Kuttruff, 2000]

$$a = 1 - |R|^2 \tag{1}$$

and thus the absorption area is

$$A = \bar{a}S \tag{2}$$

where $\bar{a}$ is the average absorption coefficient along the room boundaries.
An important room parameter that can be measured from the RIR is RT, which is the time that takes the energy in a room to decay by 60 dB once the source is turned off. By assuming that until the source was turned off it had been producing a stationary white noise, RT can be calculated from the RIR by using Schroeder's energy decay curve [Schroeder, 1965]

$$e(t) = 10\log_{10}\int_t^\infty h^2(\tau)\,\mathrm{d}\tau - 10\log_{10}\int_0^\infty h^2(\tau)\,\mathrm{d}\tau \tag{3}$$

where $h(t)$ is the RIR, and numerically solving

$$e(\mathrm{RT}) = -60\mathrm{dB}. \tag{4}$$

In the ISO 3382 standard [ISO 3382:1997, 1997], RT is calculated from a least squares based linear fitting of Schroeder's energy decay curve in order to compensate for the non-linearity and for the noise-floor effect.
Room response from a source to a receiver can be given in the frequency domain by the *room transfer function* (RTF). In rectangular rooms, the RTF is known to be a combination of

natural or *eigen* modes. At frequencies where the density of the eigenmodes is more than three eigenmodes for a 3dB bandwidth of a given eigenmode, the sound field is usually considered to sufficiently satisfy the assumptions of diffuse field theory. In diffuse fields, RT is related to the volume by Sabine formula [Kinsler et al., 2000]

$$\mathrm{RT} = 0.161\frac{V}{A}. \tag{5}$$

3. Feature extraction and normalization

A commonly used procedure of MFCC feature extraction is shown in Fig. 1 [Bimbot et al., 2004]. The pre-emphasis filter is applied to enhance the high frequencies of the spectrum, which are generally reduced by the speech production process. The STFT block splits the signal in the time domain into overlapping frames where the signal is considered to be stationary, and calculates the *fast Fourier transform* (FFT) of each frame. Then, filter banking is applied by integrating the magnitude FFT of the signal frames with triangular windows in the mel-frequency domain. Afterwards, the dB level is calculated. This results in a series of energy scalars for every frame. *Discrete cosine transform* (DCT) is calculated, from which coefficients are selected to form MFCC feature vectors. Applying a discrete-time derivative results in ΔMFCC feature vectors, such that

$$\mathbf{c}_t = [c_1^t \ldots c_N^t, \Delta c_1^t \ldots \Delta c_N^t]^{\mathrm{T}} \tag{6}$$

is the feature vector of the *t*'th frame (*t* here is a discrete time index), where *N* is the number of MFCC coefficients.

Transmission channels may add a convolutive effect to the speech signal prior to the process of feature extraction. This may result in feature vectors distortion. For that reason feature normalization may be used. In this chapter we discuss the CMS technique, which is the operation of subtracting the sample mean [Bimbot et al., 2004]

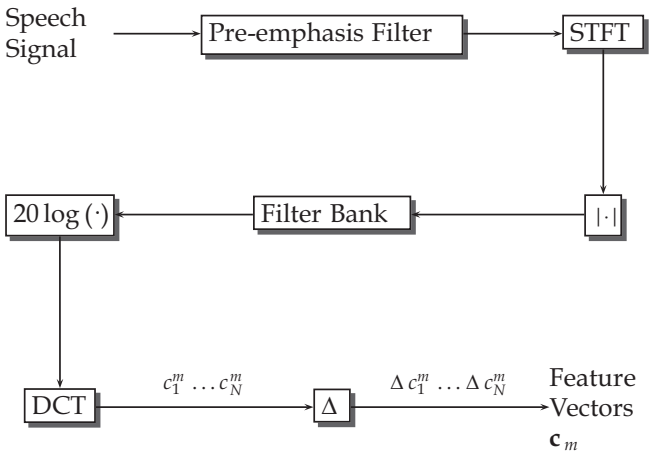


Fig. 1. Extraction of MFCC and ΔMFCC feature vectors from speech signal [Bimbot et al., 2004].

$$\tilde{\mathbf{c}}_t = \mathbf{c}_t - \boldsymbol{\mu} \qquad t = 0 \ldots T-1 \tag{7}$$

where $\boldsymbol{\mu}$ is the sample mean of the series $\mathbf{c}_0 \ldots \mathbf{c}_{T-1}$. The operation of CMS may include variance normalization [Mammone et al., 1996] by dividing the components by the sample *standard deviation* (STD), i.e.,

$$\begin{aligned} \bar{c}_n^t &= \frac{\tilde{c}_n^t}{\sigma_n} \qquad t = 0 \ldots T-1 \\ & \qquad\quad n = 1 \ldots N \end{aligned} \tag{8}$$

where for every $n = 1 \ldots N$, σ_n is the sample STD of the series $c_n^0 \ldots c_n^{T-1}$.

4. Speaker verification with GMM approach

In this section we represent a brief description on SVR with GMM approach [Bimbot et al., 2004, Mammone et al., 1996]. Speaker verification is the task of accepting or rejecting a tested speaker as a hypothetical speaker. Let

$$\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{T-1}] \tag{9}$$

be a segment of speech feature vectors $\mathbf{x}_t$ of discrete time $t \in \{0, 1, \ldots, T-1\}$. Let H_1 represent the event that the tested speaker is the hypothetical speaker, and let H_0 represent the opposite event.

The model λ_1 is defined to contain the parameters such that a parametric *probability density function* (PDF) $p(\mathbf{X}; \lambda_1)$ would model the conditional PDF $p(\mathbf{X} \mid H_1)$. In a similar way, λ_0 is defined such that $p(\mathbf{X}; \lambda_0)$ models $p(\mathbf{X} \mid H_0)$. For example, if the models assume Gaussian distribution, then λ_0 and λ_1 consist of a mean vector and a covariance matrix.

The decision is then made according to the *log-likelihood ratio test* (LLRT)

$$\Lambda(\mathbf{X}) = \log p(\mathbf{X}; \lambda_1) - \log p(\mathbf{X}; \lambda_0) \begin{cases} \geq \theta & \text{accept hypothetical speaker} \\ < \theta & \text{reject hypothetical speaker} \end{cases} \tag{10}$$

where $\Lambda(\mathbf{X})$ is referred to as the score function, and θ is the LLRT threshold. If the feature vectors in $\mathbf{X}$ are assumed independent, then for each model, $\log p(\mathbf{X}; \lambda)$ may be calculated by

$$\log p(\mathbf{X}; \lambda) = \sum_{t=0}^{T-1} \log p(\mathbf{x}_t; \lambda). \tag{11}$$

In applications where different speakers have a different number of feature vectors, the score function may be normalized by T to form

$$\tilde{\Lambda}(\mathbf{X}) = \frac{1}{T} \Lambda(\mathbf{X}), \tag{12}$$

in order not to bias the score in favor of speakers with more feature vectors.

According to the GMM approach, if $\mathbf{x}$ is a feature vector, and λ is a set of parameters, then

$$p(\mathbf{x}; \lambda) = \sum_{i=1}^{M} \omega_i p_i \left(\mathbf{x}; \lambda^{(i)} \right) \tag{13}$$

where M is the number of Gaussians in the GMM, or, the model order, the weights ω_i apply

$$\sum_{i=1}^{M} \omega_i = 1, \tag{14}$$

and $p_i(\mathbf{x}; \lambda^{(i)})$ is a parametric normal PDF. Hence, the sub-model $\lambda^{(i)}$ consists of a mean vector $\boldsymbol{\mu}_i$ and a covariance matrix $\boldsymbol{\Sigma}_i$ parameters of a single Gaussian. Hence,

$$p_i(\mathbf{x}) = p(\mathbf{x}\,; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{\frac{d}{2}} \left|\Sigma_i\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)} \tag{15}$$

where d is the dimension of $\mathbf{x}$. According to (15), the model λ in (13) can be denoted as [Reynolds et al., 2000]:

$$\lambda = \{\omega_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1\ldots M} \tag{16}$$

The parameters ω_i, $\boldsymbol{\Sigma}_i$, and $\boldsymbol{\mu}_i$ are estimated using the *expectation maximization* (EM) algorithm [Dempster et al., 1977]. The covariance matrix $\boldsymbol{\Sigma}_i$ can be selected as either *diagonal* or a *full* matrix. The interpretation of a diagonal covariance matrix is that the feature vector coordinates are independent of one another. The computation of the parametric PDFs is much simpler in this case. The advantage of the full covariance matrix, however, is the enhanced generalization of the parametric PDFs in modeling the conditional PDFs. In practice, GMM is used with diagonal covariance matrices to approximate the case of one Gaussian with a full covariance matrix with less computational effort.

Speakers that are known to a certain hypothesis are referred to as target speakers of that hypothesis, and impostor speakers to other hypotheses. Performance analysis of SVR is measured with *miss probability*, P_{MISS}, which is the probability that a target model was rejected.

$$P_{MISS} = P(\Lambda(\mathbf{X}) < \theta \mid \text{target}), \tag{17}$$

and with the *probability of false alarm*, P_{FA}, which is the probability that an impostor speaker was accepted

$$P_{FA} = P(\Lambda(\mathbf{X}) > \theta \mid \text{impostor}). \tag{18}$$

Both P_{MISS} and P_{FA} are functions of the threshold θ, and they each come at the expense of the other. The threshold θ is used as a parameter to yield the *detection error trade-off* (DET) curve, which plots P_{MISS} as a function of P_{FA}. The point on the DET curve where P_{MISS} equals P_{FA} is the EER. The EER is usually used as a scalar measure of the performance of SVR systems.

5. The effect of reverberation on the feature vectors in GMM

For reverberant speech, if the RT is larger than the *short time Fourier transform* (STFT) frame size, there will be time-smearing of the feature vectors. An increase in RT increases this time-smearing. This effect may cause the Gaussian means of the GMM to come closer together. In order to examine this, the weighted average distance between the Gaussians in the GMM and the overall mean feature vector can be calculated in the following form:

$$D = \sum_{i=1}^{M} \omega_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\mathrm{T}} (\boldsymbol{\mu}_i - \boldsymbol{\mu}), \tag{19}$$

where M is the GMM order, ω_i is the weight of the i'th Gaussian, $\boldsymbol{\mu}_i$ is the mean vector of the i'th Gaussian, and $\boldsymbol{\mu}$ is the overall mean feature vector. It is assumed that if an increase of RT results in closer Gaussians, then D should decrease.

Figure 2 shows an example of the weighted average distance between the Gaussians in GMMs that are trained from reverberant speech signals, which are the result of a convolution with simulated RIRs. A normalized form of this distance

$$D_{\mathrm{norm}} = \frac{D}{D_{\mathrm{RT=0}}} \tag{20}$$

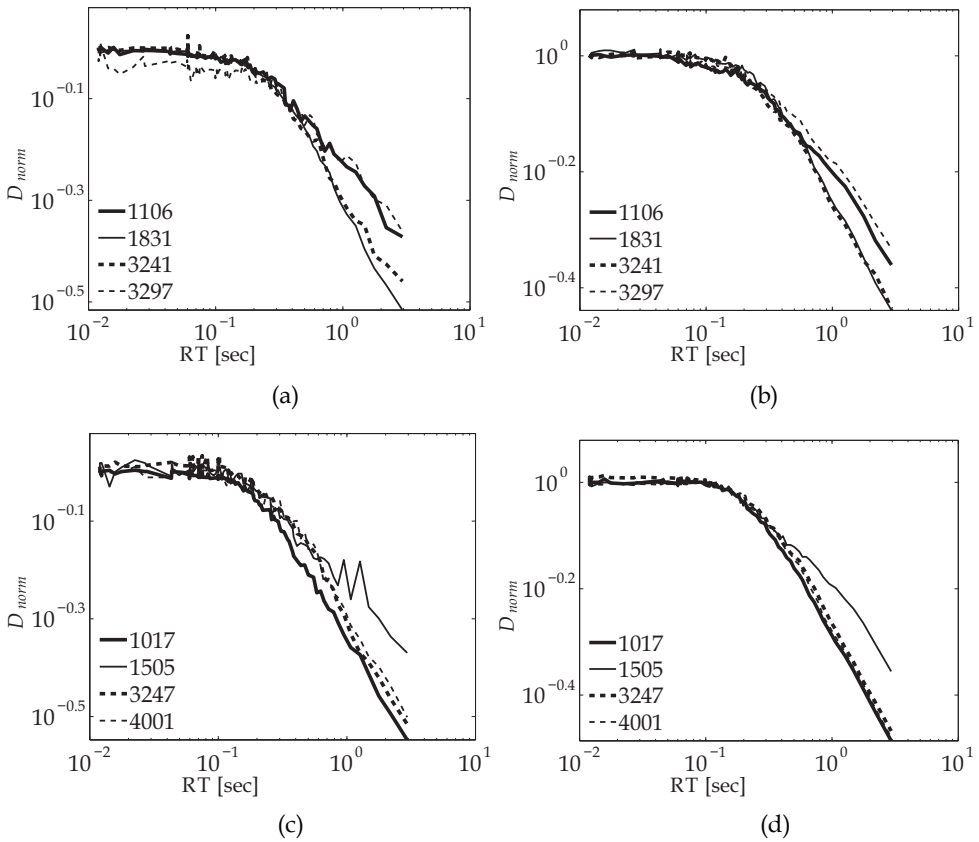


Fig. 2. Normalized distance between Gaussians and overall mean in a GMM of different speakers. Numbers in the legend indicate speaker index in NIST-SRE database. (a) 10 Gaussians, male speakers; (b) 50 Gaussians, male speakers; (c) 10 Gaussians, female speakers; and (d) 50 Gaussians, female speakers.

was used, where $D_{\mathrm{RT=0}}$ indicates the weighted average distance between the Gaussians and the overall mean feature vector in the case of clean (non-reverberant) speech. The feature space of the GMMs in Fig. 2 is of 24 dimensions, and the feature vectors consist of 12 MFCC and 12ΔMFCC coefficients. The normalized distance is displayed as a function of RT in both logarithmic axes. The STFT frame size is 30 ms. The numbers in the legend of Fig. 2 represent indices of speakers from the NIST-99 SRE database (see Sec. 7). The GMMs were trained using 10 and 50 Gaussians both for male and female speakers.
The value of D_{norm} seems to decrease with the increase of RT. Hence, the Gaussian means of the GMM come closer together. As a result, the GMM might need fewer Gaussians. Also seen from Fig. 2 is a knee between 100 and 200 msec, where RT is considerably larger than the STFT frame size. It should be pointed out that this knee applies to all speakers at similar RTs.

6. The effect of reverberation on the optimal GMM order

We aim to find an optimal GMM order for reverberant speech. The *Bayesian*, *Akaike*, and *Kullback information criteria* (BIC, AIC, and KIC, respectively) [Chen & Huang, 2005] were used to estimate the unknown order of the target models with the training observation. The criteria are defined as follows

$$\mathrm{BIC}_{\lambda_M,\mathbf{X}} = -\log p(\mathbf{X}|\lambda_M) + \frac{1}{2}M(2d+1)\log N \tag{21}$$

$$\mathrm{AIC}_{\lambda_M,\mathbf{X}} = -\log p(\mathbf{X}|\lambda_M) + M(2d+1) \tag{22}$$

$$\mathrm{KIC}_{\lambda_M,\mathbf{X}} = -\log p(\mathbf{X}|\lambda_M) + \frac{3}{2}M(2d+1) \tag{23}$$

where M is the order of the model λ_M, N is the number of feature vectors in the realization $\mathbf{X}$, and d is the feature vector dimension. The optimal model order $M^\star$ was selected for an information criterion IC, as the one whose model λ_M amongst $M \in \{10, 20, 30, 40, 50\}$ yields the minimum criterion value for $\mathbf{X}$, or,

$$M^\star_{\mathrm{IC}} = \arg \min_{M\in\{10,20,30,40,50\}} \mathrm{IC}_{\lambda_M,\mathbf{X}} \tag{24}$$

where IC is one of the information criteria defined above.
Figure 3 shows an example of the KIC and BIC values of GMMs that are trained from both non-reverberant speech signal and reverberant speech signal, which is the result of a convolution with simulated RIR. The IC values with 10, 20, 30, 40, and 50 Gaussians were normalized for each speaker with the IC value of 30 Gaussians to yield $\mathrm{IC}_{\mathrm{norm}}$. The numbers in the legend of Fig. 3(a) represent indices of speakers from the NIST-99 SRE database (see Sec. 7), and apply to all sub-figures in Fig. 3. It can be seen that optimal model order in terms of minimum KIC for clean speech is 50 ($M^\star_{\mathrm{KIC}}$ in Fig. 3(a)), whereas for reverberant speech with RT=0.85 sec (Fig. 3(b)) it reduces to some value in the range of 30 ÷ 50. Optimal model order in terms of minimum BIC is in the range of 20 ÷ 40 ($M^\star_{\mathrm{BIC}}$ in Fig. 3(c)), whereas for reverberant speech with RT=0.85 sec (Fig. 3(d)) it reduces to 10. The general effect of

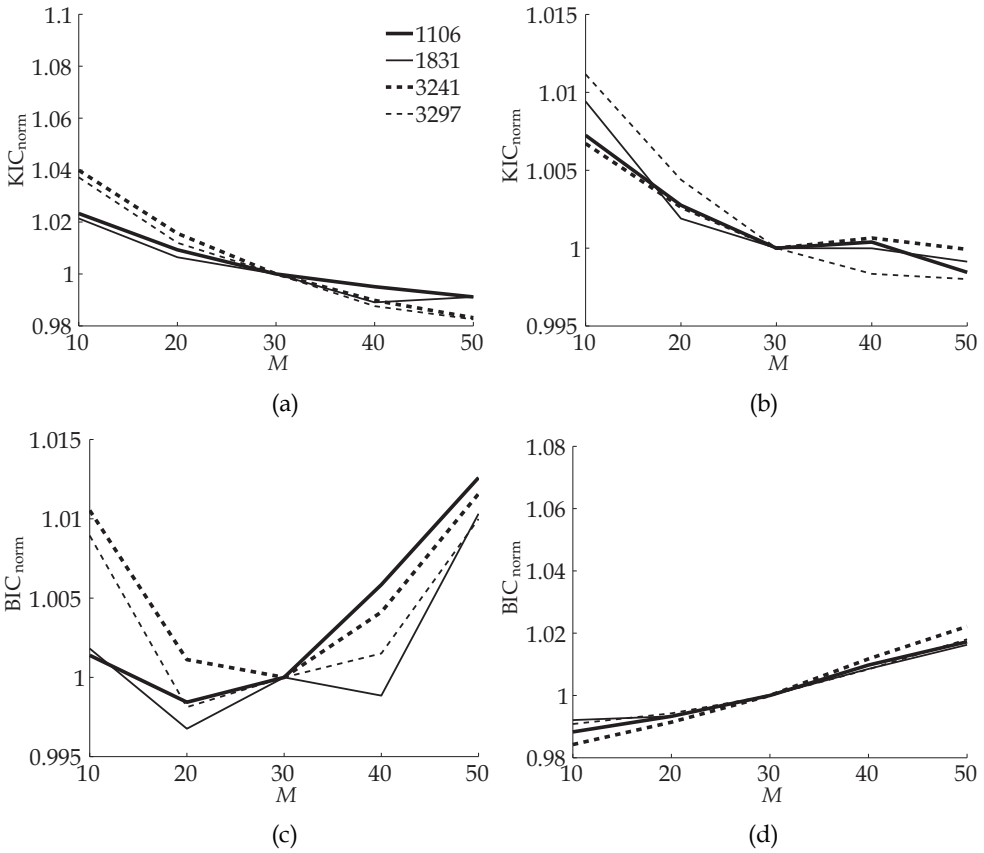
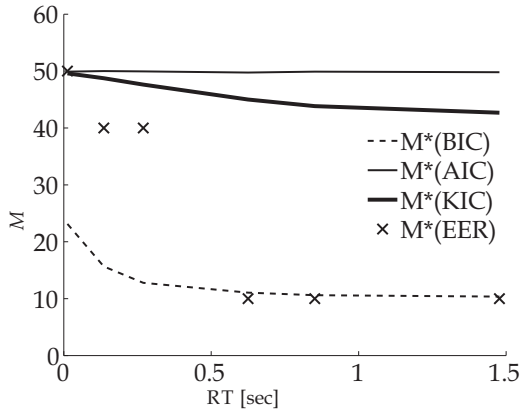


Fig. 3. KIC and BIC values as a function of GMM order (normalized with KIC and BIC values in case of 30 Gaussians), using clean and reverberant speech of male speakers. (a) KIC without reverberation, (b) KIC with RT=0.85 sec, (c) BIC without reverberation, and (d) BIC with RT=0.85 sec.

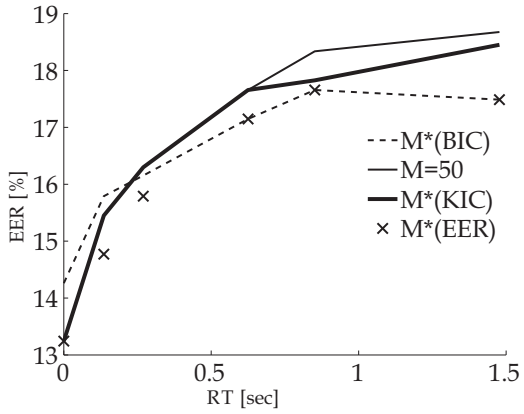
reverberation is therefore to reduce the optimal model order. It should be noted, however, that the results have a large variance of model order, but a low variance of IC values. Therefore, the significance of model order should be examined in terms of minimum EER of a SVR system.

7. Experimental study of the effect of GMM order on SVR

In this section, an experimental study of the effect of GMM order on the EER of SVR is presented. Reverberant speech training data were generated for several values of RT. The image method of Allen and Barkley [Allen & Berkley, 1979] was used to generate a simulated impulse response of a room. RT is measured on the impulse response according to [Schroeder, 1965]. Speech segments were taken from the *national institute of standards and technology* (NIST) – 1999 *speaker recognition evaluation* (SRE) database [Martin and Przybocki, 2000] for training target GMMs.



(a)



(b)

Fig. 4. Comparing results of optimal GMM order using BIC, AIC, and KIC, to the optimal GMM order in terms of minimal EER. (a) Optimal model order, (b) EER values.

Figure 4(a) shows the optimal order for an average of 198 male speakers with one-minute long speech segment each. Figure 4(a) also compares the optimal order of BIC, AIC, and KIC to the optimal order in terms of minimum EER (the model order among $M \in \{10, 20, 30, 40, 50\}$ that yields the minimum EER). EER results were generated by a loglikelihood based SVR experiment. This experiment included 686 half-minute long reverberant test speech segments, generated from the NIST- 99 SRE database. The test speech was introduced to the reverberant target GMMs and to a background GMM. The Background GMM was generated from 50 speakers, each with a one-minute long speech segment, taken from the

NIST-98 SRE database, using reverberant speech with the same RT of the test speech, and a constant model order of 256 Gaussians. No channel compensation was used.

It can be seen that the optimal model order is reduced with the increase of RT for model selection according to BIC and KIC. For RT < 0.5 sec, $M^{\star}_{\mathrm{KIC}}$ is similar to the optimal model order in terms of minimum EER. For RT > 0.5 sec, $M^{\star}_{\mathrm{BIC}}$ is similar to the optimal model order in terms of minimum EER. $M^{\star}_{\mathrm{AIC}}$ is constant 50.

Fig. 4(b) shows the EER results for the optimal model order of KIC and BIC, compared with EER of a constant model order 50, and to the minimum EER. It can be seen that in terms of EER values, using a constant model order 50 is similar to using $M^{\star}_{\mathrm{KIC}}$ for RT < 0.5 sec, but worse than using $M^{\star}_{\mathrm{BIC}}$ for RT > 0.5 sec. Since $M^{\star}_{\mathrm{BIC}}$ decreases with the increase of RT, reducing model order can reduce the EER of SVR in a highly reverberant environment.

8. Experimental study on the performance of CMS applied in SVR under reverberation

An early study of the authors [Shabtai et al., 2008b] has investigated the effect of reverberation on the efficiency of CMS in improving the performance of SVR. The performance of an SVR system was measured by calculating the EER in rooms with different RTs and volumes. Test speech segments were made reverberant with RIRs that were simulated using the image method of Allen and Barkley [Allen & Berkley, 1979]. It was shown that for high RTs, the efficiency of CMS decreases.

In this section we extend the research to reverberant speech generated by convolution with measured RIRs. The environments in which the RIRs were measured are tabulated in Tab. 1. Measured RIRs 1 ÷ 10 were measured with Brüel & Kjær 4295 Omni-Source loudspeaker and Brüel & Kjær 4942 $\frac{1}{2}$-inch diffuse-field microphone, at selected rooms in *Ben-Gurion University of the Negev, Israel* (BGU). Measured RIRs 11 ÷ 14 were taken from the *Concert Hall Research Group* (CHRG) project [CHR, 2004]. In order to compare the results with simulated RIRs, the image method was used to simulate RIRs of rooms with similar dimensions and RTs to the rooms in Tab. 1.

The SVR system was using 20 msec speech frames in which MFCC and ΔMFCC were calculated to form 24-dimensional feature vectors, for which CMS was either applied or not. Target models were trained using the AGMM approach [Reynolds et al., 2000]. A *background GMM* (BGM) of 1024 Gaussians was generated from one-minute long non-reverberant speech segments of 50 speakers, taken from the NIST- 1998 SRE database. This BGM was used to train target AGMMs for 198 male speakers, with one-minute long non-reverberant speech segments, taken from the NIST-1999 SRE [Martin and Przybocki, 2000] database. Test speech segments were taken from NIST-1999 SRE, for 686 male speakers with half-minute long speech segment each. The test speech segments were made reverberant by convolving them with simulated and measured RIRs. The EER results were calculated by introducing the reverberant test speech segments to the target AGMMs and BGM of non-reverberant speech.

Figure 5 shows a scatter plot of EER values as a function of RT. The cross, circle, and triangle marks on Fig. 5 represent EER values when either feature normalization was not used, or CMS was applied, or CMS was applied along with variance normalization, respectively. Linear fitting to the EER values is shown in Fig. 5. Thick solid curves denote using no

RIR	Environment	RT [sec]	V_i[m^3]
1	Building 33 Office 126	0.8	37
2	Building 33 Office 427	0.6	42
3	Building 34 Classroom 103	0.6	120
4	Building 33 Lecture room 102	0.5	147
5	Building 34 Classroom 202	1	301
6	Building 33 Teaching lab 204	0.6	339
7	Building 26 Auditorium 4	1.5	793
8	Building 26 Auditorium 5	1.2	1142
9	Building 26 Auditorium 6	1.3	1142
10	Sonnenfeld lecture room	1	2529
11	Mechanics Hall (Worchester, MA)	2.4	8367
12	Troy Music Hall (Troy, NY)	2.6	11320
13	Boston Symphony Hall (Boston, MA)	2.6	16611
14	Kleinhans Music Hall (Buffalo, NY)	1.9	18241

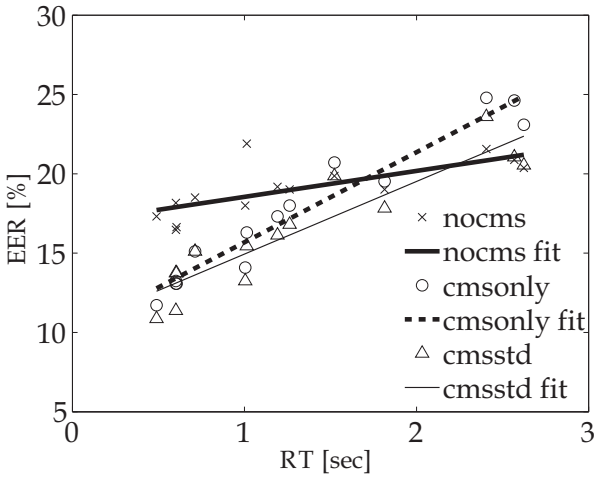
Table 1. Rooms in which RIRs were measured.

feature normalization, dashed curves denote using CMS, and with thin solid curves denote using CMS along with variance normalization. Figures 5(a) and 5(b) refer to simulated and measured RIRs, respectively. In the case of simulated RIRs as well as in the case of measured RIRs, it can be seen that CMS is improving the performance of SVR in a reduced manner with the increase of RT. Moreover, it can be seen that for some high values of RT, CMS may increase the EER rather than decrease it. These results support previous results [Shabtai et al., 2008b] in which it was shown that CMS is improving the performance of SVR in a reduced manner with the increase of RT, and validate them with measured RIRs.

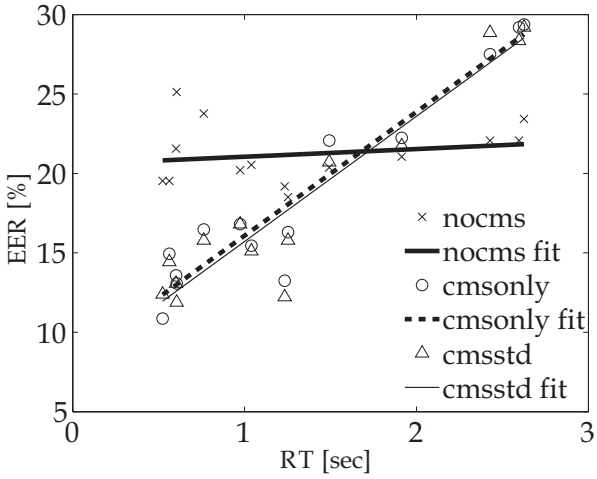
9. Conclusion

The effect of GMM order on SVR with reverberant speech was investigated. Time-smearing of the feature vectors due to reverberation reduces the optimal GMM order in terms of minimum BIC and KIC. When tested on a GMM-based SVR system, reducing model order improves system performance for highly reverberant speech. A future adjustment to AGMM may be proposed in this direction

The effect of room volume and RT on the performance of CMS applied to MFCC feature vectors in SVR was investigated. It was shown that the performance of CMS may degrade with the increase of RT. In some cases of high RT, CMS may increase the EER of SVR rather than decrease it. Hence, in these cases, CMS should not automatically be used. As a future work, we purpose combining a CMS decision block in SVR.



(a)



(b)

Fig. 5. EER values of SVR with reverberant speech as a function of RT. Cross marks ("x") denote no feature normalization (linear fitting with thick solid line), circles ("o") denote CMS (linear fitting with thick dashed line), and triangles ('Δ') denote using CMS with variance normalization (linear fitting with thin solid line). Test speech segments were made reverberant by convolution with (a) simulated, and (b) measured RIRs.

10. References

[CHR, 2004] (2004). *Concert Hall Research Group CD v.3*. Concert Hall Research Group, 327F Boston Post Road. Sudbury, MA 01776. Attention: Timothy J. Foulkes, email: chrg@cavtocci.com, phone: 978.443.7871, fax: 978.443.7873.

[Allen and Berkley, 1979] Allen, J. B. and Berkley, D. A. (1979). Image method for efficiently simulating small room acoustics. *J. Acoust. Soc. Am.*, 65(4):943–950.

[Bimbot et al., 2004] Bimbot, F., Bonastre, J. F., Fredouille, C., Gravier, G., Meignier, S., Merlin, T., Garcia, J. O., Chagnolleau, I. M., Delacretaz, D. P., and Reynolds, D. A. (2004). A tutorial on text-independent speaker verification. *EURASIP J. Appl. Signal Processing*, 2004(4):430–451.

[Chen and Bilmes, 2007] Chen, C. P. and Bilmes, J. A. (2007). MVA processing of speech features. *IEEE Trans. Speech Audio Process.*, 15(1):257–270.

[Chen and Huang, 2005] Chen, H. and Huang, S. (2005). A comperative study on model selection and multiple model fusion. In *Proc. FUSION*, volume 1, pages 820–826.

[Davis and Mermelstein, 1980] Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-28(4):357–366.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc.*, 39(1):1–38.

[ISO 3382:1997, 1997] ISO 3382:1997 (1997). Acoustics – measurement of the reverberation time of rooms with reference to other acoustical parameters.

[Kinsler et al., 2000] Kinsler, L. E., Frey, A. R., Coppens, A. B., and Sanders, J. V. (2000). *Fundamentals of Acoustics*. John Wiley, New York.

[Kuttruff, 2000] Kuttruff, H. (2000). *Room Acoustics*. Spon Press, New York.

[Mammone et al., 1996] Mammone, R. J., Zhang, X., and Ramachandran, R. P. (1996). Robust speaker recognition: a feature-based approach. *IEEE Signal Process. Mag.*, 13(5):58–71.

[Martin and Przybocki, 2000] Martin, A. and Przybocki, M. (2000). The NIST 1999 speaker recognition evaluation - an overview. *Digital Signal Processing*, 10(1–3):1–18.

[Peer et al., 2008] Peer, I., Rafaely, B., and Zigel, Y. (2008). Reverberation matching for speaker recognition. In *Proc. ICASSP*, pages 4829– 4832.

[Reynolds et al., 2000] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1):19–41.

[Schroeder, 1965] Schroeder, M. R. (1965). New method of measuring the reverberation time. *J. Acoust. Soc. Am.*, 37(3):409–412.

[Shabtai et al., 2008a] Shabtai, N. R., Rafaely, B., and Zigel, Y. (2008a). The effect of GMM order and CMS on speaker recognition with reverberant speech. In *Proc. HSCMA*, pages 144–147.

[Shabtai et al., 2008b] Shabtai, N. R., Rafaely, B., and Zigel, Y. (2008b). The effect of room parameters on speaker verification using reverberant speech. In *Proc. IEEEI*, pages 231–235.

[Zigel and Wasserblat, 2006] Zigel, Y. and Wasserblat, M. (2006). How to deal with multiple targets in speaker identification systems? In *Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, pages 1–7.

4

Body-Conducted Speech Recognition and its Application to Speech Support System

Shunsuke Ishimitsu
Hiroshima City University
Japan

1. Introduction

In recent years, speech recognition systems have been used in a wide variety of environments, including internal automobile systems. Speech recognition plays a major role in a dialogue-type marine engine operation support system currently under investigation. In this system, speech recognition would come from the engine room, which contains the engine apparatus, electric generator, and other equipment. Control support would also be performed within the engine room, which means that operations with a 0-dB signal-to-noise ratio (SNR) or less are required. Noise has been determined to be a portion of speech in such low SNR environments, and speech recognition rates have been remarkably low. This has prevented the introduction of recognition systems, and up till now, almost no research has been performed on speech recognition systems that operate in low SNR environments. In this chapter, we investigate a recognition system that uses body-conducted speech, that is, types of speech that are conducted within a physical body, rather than speech signals themselves. Since noise is not introduced into body-conducted signals that are conducted in solids, even within sites such as engine rooms which are low SNR environments, it is necessary to construct a system with a high speech recognition rate. However, when constructing such systems, learning data consisting of sentences that must be read a number of times is required for creation of a dictionary specialized for body-conducted speech. In the present study we applied a method in which the specific nature of body-conducted speech is reflected within an existing speech recognition system with a small number of vocalizations.

On the other hand, people with speech disabilities face communication problems in daily conversation. They can communicate with substitute speech, but this does not have the required frequency to be readily understood in daily conversation. Therefore, we have proposed the speech support system with body-conducted speech recognition. The system retrieves speech from the body-conducted speech via a transfer function using recognition to decide on a subword sequence and the duration. Before constructing the system, we examined the effectiveness of body-conducted speech recognition for communication disorders. The first step in constructing the system involved investigating continuous word unit speech recognition, using an acoustic model not suited to body-conducted speech for communication disorders. In this study, we analyzed each parameter of these speeches and experimented with body-conducted speech recognition. We concluded that an adaptation using body-conducted speech recognition to achieve high recognition performance for disorders is valid.

2. Noise-robust body-conducted speech recognition system

2.1 Dialogue-type marine engine operation support system using body-conducted speech

Since the number of sailors has decreased dramatically in recent years, there is a shortage of skilled maritime engineers. Therefore, a database which stores the knowledge used by skilled engineers has been constructed (Matsushita & Nagao,2001).

In this study, this knowledge database is accessed by speech recognition. The system can be used to educate sailors and make it possible to check the ship's engines.

Figure 1 shows a conceptual diagram of a dialogue-type marine engine operation support system using body-conducted speech. The signals are detected with a body-conducted microphone and then wirelessly transmitted, and commands or questions from the speech-recognition system located in the engine control room are interpreted. A search is made for a response to these commands or questions speech recognition results and confirmation on the suitability of entering such commnads into the control system is made. Commands suitable for entry into the control system are speech-synthesized and output to a monitor. The speech-synthesized sounds are replayed in an ear protector/speaker unit, and while continuing communication, work can be performed while safety is continuously confirmed. The present research is concerned with the development of the body-conducted speech recognition portion of this system. In this portion of the study, a system was created based on a recognition engine that is itself based on a Hidden Markov Model (HMM) incidental to a database (Itabashi, 1991).

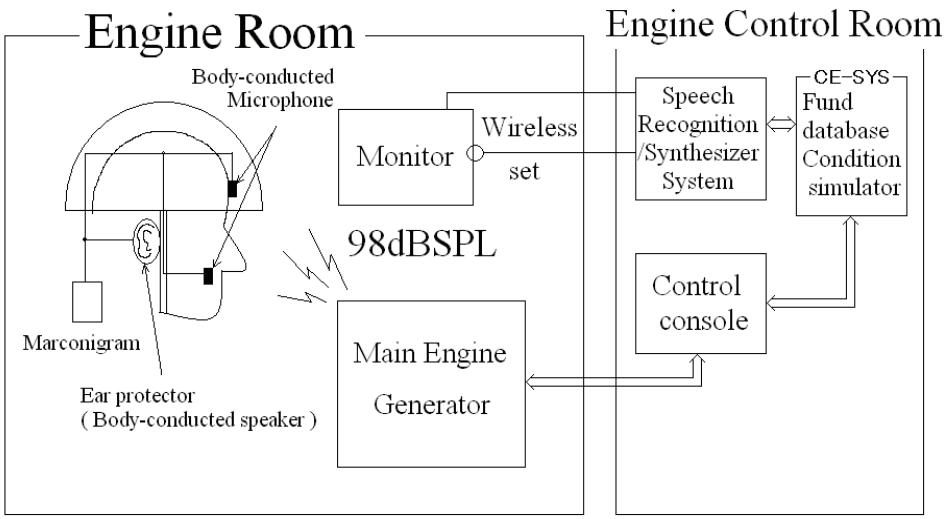


Fig. 1. Dialogue-type marine engine operation support system using body-conducted speech.

With this system, multivariate normal distribution is used as the output probability density function, and a mean vector μ that takes an n-dimensional vector as the frame unit of speech feature quantities and a covariance matrix Σ are used; these are expressed as follows: (Baum, 1970)

$$b(o,\mu,\Sigma)=\frac{1}{(2\pi)^{n/2}\,|\,\Sigma\,|^{1/2}}e^{-\frac{1}{2}(o-\mu)^t\Sigma^{-1}(o-\mu)} \quad (1)$$

HMM parameters are shown using the two parameters of this output probability and the state transition probability. To update these parameters using conventional methods, utterances repeated at least 10-20 times would be required. To perform learning with only a few utterances, we focused on the relearning of the mean vector μ within the output probability, and thus created a user-friendly system for performing adaptive processing.

2.2 Investigation into identifying sampling locations for body-conducted speech
2.2.1 Investigation through frequency characteristics

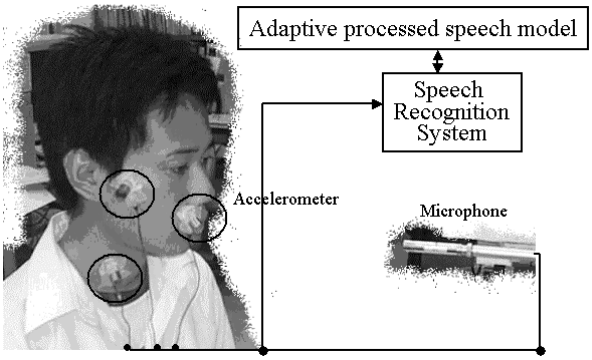


Fig. 2. Sampling location for body-conducted speech.

Figure 2 shows candidate locations for body-conducted speech during this experiment. Three locations - the lower part of the pharynx, the upper left part of the upper lip and the front part of the zygomatic arch - were selected as signal sampling locations. The lower part of the pharynx is an effective location for extracting the fundamental frequency of a voice and is often selected by electroglottograph (EGG). Since the front part of the zygomatic arch is where a ship's chief engineer has his helmet strapped to his chin, it is a meaningful location for sound-transmitting equipment. The upper left part of the upper lip is the location that was chosen by Pioneer Co., Ltd. for application of a telecommunication system in a noisy environment. This location is confirmed to have very high voice clarity (Saito et al., 2001). Figure 3 indicates the amplitude characteristics of body-conducted speech signals at each location, and also shows the difference between a body-conducted signal on the upper lip and the voice when a 20-year-old male reads "Denshikyo Chimei 100" (this is the Japan Electronics and Information Technology Industries Association (JEITA) Data Base selection of 100 locality names). Tiny accelerometers were mounted on the above-mentioned locations with medical tape. Figure 3 indicates that the amplitudes of body-conducted speech at the zygomatic arch and the pharynx are 10-20 dB lower than body-conducted speech at the upper left part of the upper lip. The clarity of vibration signals from body-conducted speech was poorer using signals from all sites except the upper left part of the upper lift in the listening experiment. Some consonant sounds that were not captured at other locations were extracted at the upper left part of the upper lip. However, compared to

the speech signals shown in Figure 4, the amplitude characteristics at the upper left part of the upper lip appear to be about 10 dB lower than those of the voice. Based on frequency characteristics, we believe that recognition of a body-conducted signal will be difficult utilizing an acoustic model built using acoustic speech signals. However, by using the upper left part of the upper lip, the site with the highest clarity signals, we think it will be possible to recognize body-conducted speech with an acoustic model built from acoustic speech using adaptive signal processing or speaker adaptation.

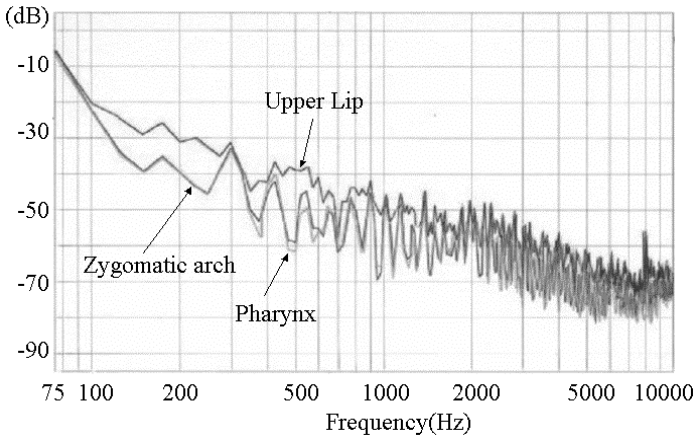


Fig. 3. Frequency characteristics of body-conducted speech.

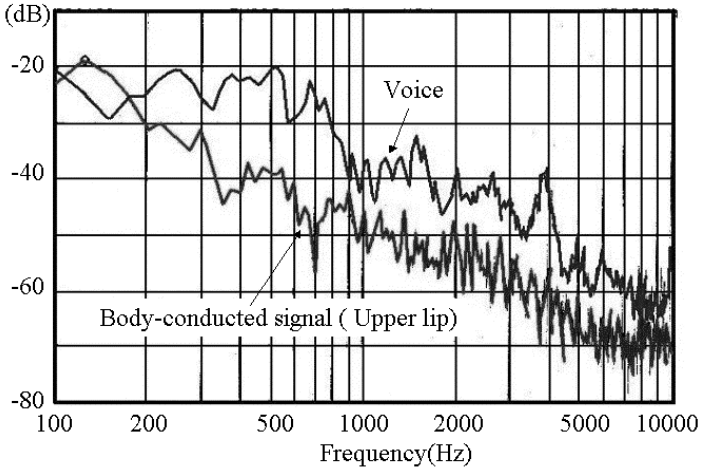


Fig. 4. Frequency characteristics of body-conducted speech and speech.

In this study, we examined a word recognition system. To investigate the possibility of building a body-conducted speech recognition system with a speech model without building an entirely new body-conducted speech model, we compared sampling locations for body-conducted speech parameters at each location, and parameter differences amongst

words. Figure 5 shows the difference on mel-cepstrum between speech and body-conducted speech at all frame averages. Body-conducted speech concentrates energy at low frequencies so that it converges on energy at lower orders like the lower part of the pharynx and the zygomatic arch, while the mel-cepstrum of signals from the upper left part of the upper lip shows a resemblance to the mel-cepstrum of speech. They have robust values at the seventh, ninth and eleventh orders and exhibit the outward form of the frequency property unevenly.

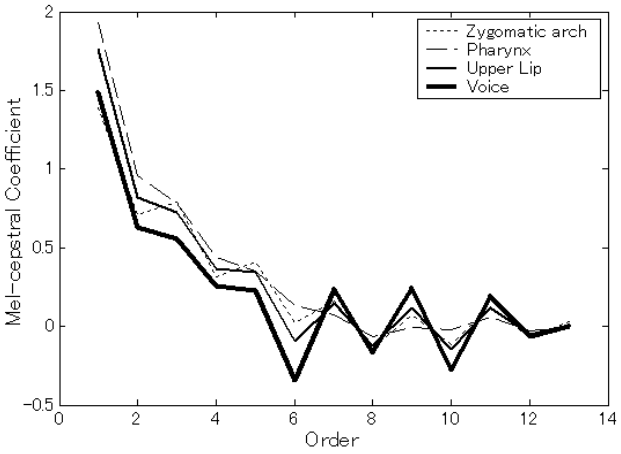


Fig. 5. Mel-cepstrum difference between speech and body-conducted speech.

Although the upper left part of the upper lip has the closest proximity to voice characteristics, it does not capture all of the characteristics of speech. This caused us to conclude that it is difficult to build a body-conducted speech model solely with a voice model.

We concluded that it might be possible to build a body-conducted speech recognition system by building a model at the upper left part of the upper lip and optimizing speech-conducted speech signals based on a voice model.

2.3 Recognition experiments

2.3.1 Selection of the optimal model

The experimental conditions are shown in Table 1. For system evaluation, we used speech extracted in the following four environments:

- Speech within an otherwise silent room
- Body-conducted speech within an otherwise silent room
- Speech within the engine room of the Oshima-maru while the ship was running
- Body-conducted speech within the engine room of the Oshima-maru while the ship was running

Noise within the engine room of the Oshima-maru when the ship was running was 98 dB SPL (Sound Pressure Level), and the SNR when a microphone was used was -25 dB. This data consisted of 100 terms read by a male aged 20, and the terms were read three times in each environment.

Valuation method	Three set utterance of 100 words
Vocabulary	JEITA 100 locality names
Microphone position	From the month to about 20cm
Accelerator position	The upper left part of the upper lip

Table 1. Experimental conditions

	anchorage		cruising	
	Speech	Body	Speech	Body
Anechoic room	45%	14%	2%	45%
Anechoic room + noise	64%	10%	0%	49%
Cabin	35%	9%	1%	42%
Cabin + noise	62%	4%	0%	48%

Table 2. The result of preliminary testing

Extractions from the upper left part of the upper lip were used for the body-conducted speech since the effectiveness of these signals was confirmed in previous research (Ishimitsu et al, 2001, Haramoto et al, 2001). the effectiveness of which has been confirmed in previous research. The initial dictionary model to be used for learning was a model for an unspecified speaker created by adding noise to speech extracted within an anechoic room. This model for an unspecified speaker was selected through preliminary testing. The result of preliminary testing is shown in Table 2.

2.3.2 The effect of adaptation processing

The speech recognition test results in the cases where adaptive processing (Ishimitsu & Fujita, 1998) was performed for room interior speech and engine-room interior speech are shown in Table 3, and in Figures 6 and 7. The underlined portions show the results of the tests performed in each stated environment. In tests of recognition and signal adaptation via speech within the machine room, there was almost no operation whatsoever. That result is shown in Figure 6, and it is thought that extraction of speech features failed because the engine room noise was louder than the speech sounds. Conversely, with room interior speech, signal adaptation was achieved. When environments for performing signal adaptation and recognition were equivalent, an improvement in the recognition rate of 27.66% was achieved, as shown in Figure 7. There was also a 12.99% improvement in the recognition rate for body-conducted speech within the room interior. However, since that recognition rate was around 20% it would be unable to withstand practical use. Nevertheless, based on these results, we found that using this method enabled recognition rates exceeding 90% with just one iteration of the learning samples.

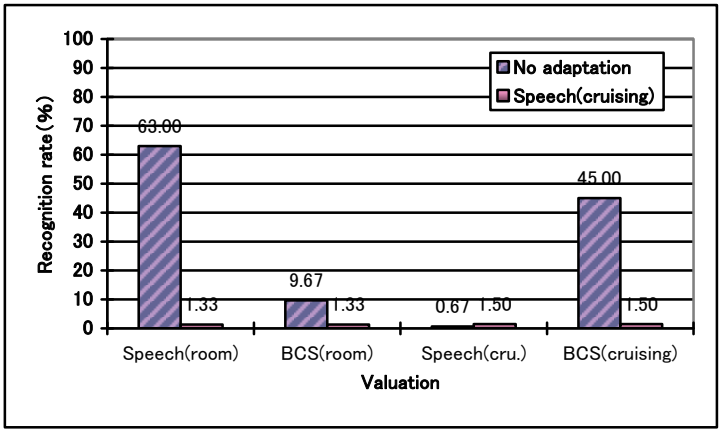


Fig. 6. Signal adaptation with speech (crusing).

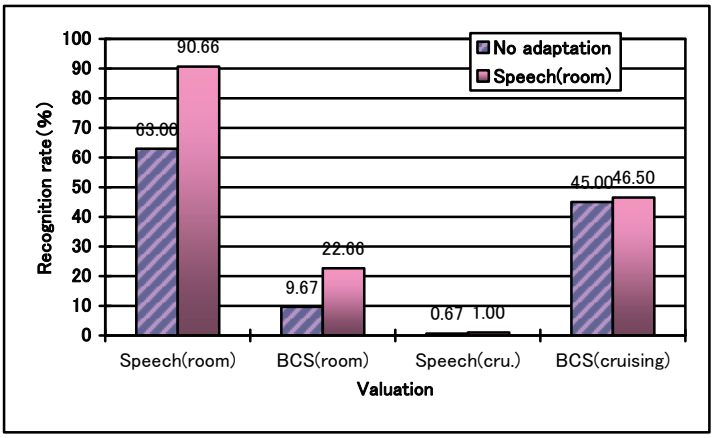


Fig. 7. Signal adaptation with speech (room).

Valuation	Candidate for adaptation		
	Room	Engine Room	No adaptation
Speech(Room)	**90.66**	1.33	63.00
Body(Room)	22.66	1.33	9.67
Speech(Engine)	1.00	**1.50**	0.67
Body(Engine)	46.50	1.50	45.00

Table 3. Result of adaptation processing with speech (%)

The results of cases where adaptive processing was performed for room-interior body-conducted speech and engine-room interior body-conducted speech are shown in Table 4,

and in Figures 8 and 9. Similar to the case where adaptive processing was performed using speech, when the environment where adaptive processing and the environment where

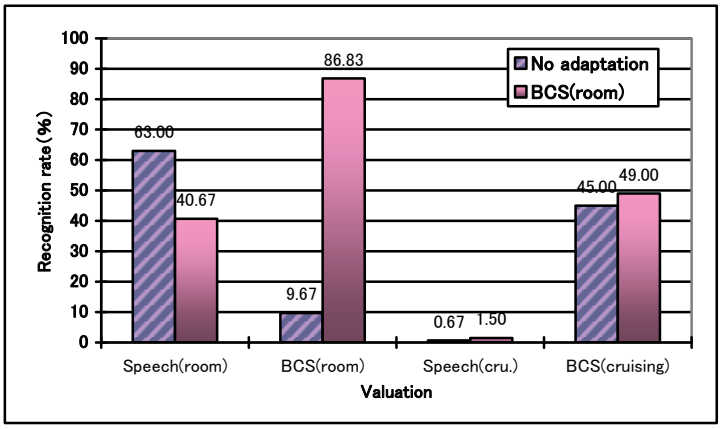


Fig. 8. Signal adaptation with body-conducted speech (room).

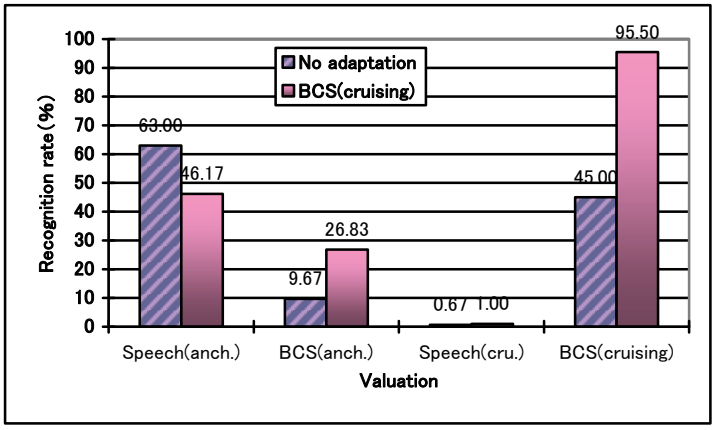


Fig. 9. Signal adaptation with body-conducted speech (crusing).

Valuation	Candidate for adaptation		
	Room	Engine Room	No adaptation
Speech(Room)	40.67	46.17	63.00
Body(Room)	**86.83**	26.83	9.67
Speech(Engine)	1.50	1.00	0.67
Body(Engine)	49.00	**95.50**	45.00

Table 4. Result of adaptation processing with body-conducted speech (%)

recognition was performed were equivalent, high recognition rates of around 90% were obtained, as shown in Figure 8. In Figure 9. It can be observed that signal adaptation using engine-room interior body-conducted speech and speech recognition results were 95% and above, with 50% and above improvements, and that we had attained the level needed for practical usage.

3. Speech support system using body-conducted speech recognition for disorders

In late year, the number of people with disabilities that impede normal speech communication has recently increased. Pharyngeal cancer is one of the many disorders affecting such people confirmed by the increasing number of pharynx-related surgery. Although most affected patients recover well after surgery, they develop speech disorders. As a result, they have to deal with speech communication problems in their daily conversations.

The most common solution used for speech disorders is esophagus vocalization, which is inexpensive and does not require surgery. Such vocalization involves inhaling air into the stomach, and then breathing it out into the surrounding air. The new glottis in the lower pharyngeal mucous membrane then vibrates, changing air into esophageal speech through the articulation organ between the pharynx and mouth. In this way, a functionally disordered individual can generate esophageal speech. However, esophageal speech does not provide optimal fundamental frequency, high-frequency component, and power for daily conversations. Therefore, people with esophagus vocalization still have problems of communication in noisy situations encountered in daily life. Many researchers have attempted to improve the quality of esophageal speech and have looked at methods to achieve clear vocalization from body-conducted speech and the construction of speech synthesis systems. Here, we describe relevant prior research for retrieving good quality esophageal speech.

Akimoto, et al. are improved its quality retrieval on fundamental frequency (Akimoto et al., 2002). Nakamura, et al. are constructed voice conversion system using transmitted artificial speech (Nakamura et al., 2007). Ando, et al. proposed speech synthesis system for Chinese language training system (Ando & Takagi, 2007). We propose speech support system using body-conducted speech recognition for disorders. This system is able to extract a signal in a noisy environment using an accelerator.

However, conventional techniques cannot create clear speech, including the speaker's particular speech characteristics. To resolve this problem, we use continuous sub-word body-conducted speech recognition and a sub-word unit transfer function database. We propose a new solution for disorders based on a speech support system that uses bodyconducted speech recognition. Typically, the system uses body-conducted speech as the vocal chord signals, so it differs from that using the vocal chords with an impulse response to the input signal (Fukushima & Kido, 2007, Morise et al., 2007).

3.1 Proposed system

Here, we describe the speech support system using body-conducted speech recognition and sub-word transfer functions. Figure 10 shows an outline of the speech support system for disorders.

First, a disabled person makes an utterance through esophageal speech, and the system extracts body-conducted speech with an accelerator pickup. Second, the system estimates

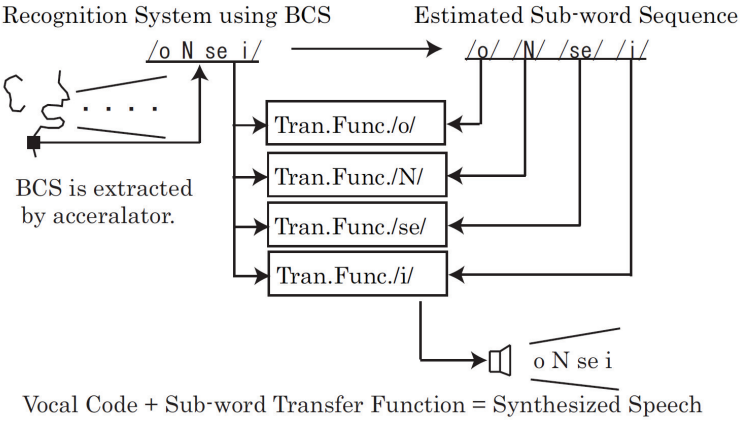


Fig. 10. Speech support system for disorders.

the sub-word unit sequence and its duration. Esophageal speech is then changed into recovery speech using the transfer function of the presumed sub-word unit through recognition of the output information. Finally, the system connects each recovery signal of the sub-word unit, and recreates the utterance with them.

This system has several advantages. Esophageal speech does not have sufficient volume compared with normal speech, and therefore, a speech disabled person faces a variety of problems in conversations with typical everyday noise. This becomes a problem when the conversation partner cannot hear the esophageal speech. However, with our system, even in a noisy environment, esophageal speech can be heard using body-conducted speech. Because the transfer function used by our system expresses each speaker's characteristics, the proposed system becomes a refection of each speaker. As well, because body-conducted speech is used as vocal cord signals, the signals hold linguistic informations such as fundamental frequency. When body-conducted speech is used, it is expected that the recovered speech will contain recognition errors and the system can then choose different transfer functions.

3.2.1 Advantages of the system

The system has following several advantages.

- The system works on high noisy environment
- Transfer functions has possess a robust individuality of each disorders characteristics
- The system uses vocal code user's body-conducted speech
- It is expected that the retrieved speech can approximate clear speech when recognition errors are considered.

Esophageal speech does not have sufficient volume compared with normal speech, so disabled people have a problem when conversing in noisy environments. However, this problem can be solved using body-conducted speech, since the signal can function correctly in noisy environments. Transfer functions in the system each express the individual characteristics of a user. The reason for this is explained in the next section. Moreover, using body-conducted speech as vocal chord information means that it contains linguistic information, such as the fundamental frequency and so on. Also, the recognition system can be amended when the system retrieves speech using a different transfer function.

3.2.2 Controversial issues in constructing the system

To construct the system, it has to examine following kinds.

- Effectiveness of continuous sub-word unit recognition system.
- Construction of continuous sub-word unit cross spectrum transfer function database.
- Effectiveness of the retrieved speech with respect to the frequency component and the ability to hear it.

Here, we discuss the effectiveness of the system for healthy people only. As a next step, we will construct a system for the speech disabled, which, as such, is beyond the scope of this paper.

3.3 Continuous sub-word recognition

3.3.1 Decoding algorithm of continuous sub-word recognition

Continuous sub-word unit recognition is important for body-conducted speech recognition in the system, since it is necessary to estimate each sub-word sequence and the duration times. This decoding system, constructed using the Julian/Julius tools, is known as Japanese Large Vocabulary Continuous Speech Recognition (LVCSR) (Kawahara et al., 1999). Although the Julius speech recognition engine needs a language model, our decoding system does not. Instead of a language model, our system contains a descriptive grammar. The continuous sub-word unit recognition includes the grammar, and is executed iteratively by a sound model and silent model of the mora or syllable unit. The decoding system is involved in sub-word continuous recognition. We have already demonstrated the effectiveness of body-conducted speech recognition using an acoustic model with the parameters estimated by body-conducted speech. By using this technique, the recognition system using body-conducted speech can correctly estimate a sub-word sequence and its duration.

3.3.2 Determination of signal sampling location for body-conducted speech

In a previous section, we examined signal sampling locations for body-conducted speech by comparing recognition parameters for each location. For this experiment, the upper lip was chosen as the signal sampling location for body-conducted speech. In the system, we use the pharynx as the body-conducted signal sampling location. This position is very close to the pharynx, so we expect this to be a suitable location for body-conducted speech as vocal code. If this sampling location is not suitable for executing this system, we will use the upper lip. The upper lip and pharynx have already been used effectively in isolated word recognition systems using body-conducted speech.

3.4 Construction of sub-word unit transfer function database

In this section, first, we explain fundamental transfer function between speech and body-conducted speech. Then we consider a transfer function between speech and body-conducted speech. We examine the word unit transfer function using a cross spectrum method as in previous research, however, this result is not effective since a word contains several consonants, and is complex compared with a sub-word. So we need to examine the effectiveness of several sub-word units of the retrieved speech, such as the syllable, semi-syllable and mora.

3.4.1 Relationship of transfer functions

Speech is synthesized by the vocal chords and the transfer function expressed by the oral and nasal cavities, while body-conducted speech is expressed by the body and skin. There is a

relationship between the transfer functions of speech and body-conducted signals as shown in Figure 11, where disabled people are those with disorders from cancer of the pharynx, and healthy people are those that are able to utter spoken speech. The Esophagus and BCS are the utterance styles for each group, respectively. BCS means body-conducted speech while Esophageal denotes esophageal speech. In this study, we propose sub-word transfer functions that allow those using body-conducted speech to speak as healthy individuals. These transfer functions are estimated using a cross spectrum method where each signal is a sub-word.

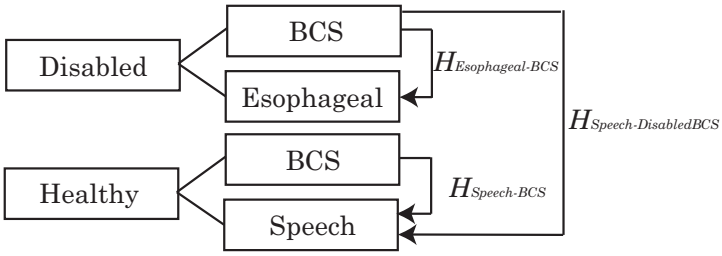


Fig. 11. Relationships of transfer functions between speech and body-conducted speech

3.4.2 Cross spectrum method

In this section, first, we will explain the basic principles of the transfer function between normal speech and body-conducted speech. Second, we describe the technique of making a sub-word unit transfer function using a cross-spectral method that makes use of speech and body-conducted speech healthy. In a previous study, we developed a word unit transfer function that used a cross-spectral method. Therefore, we investigated the validity of speech recovery with several sub-word units such as the syllable, semi-syllable, and Mora. Speech consists of a transfer function expressed as vocal cord signals, in the mouth and the nasal cavity. Moreover, as for body-conducted speech, the signals involve the body or skin. Figure 11 shows the relationship between the transfer function in speech and body-conducted speech. For every speaker, the utterance styles can be body-conducted speech body-conducted speech and esophageal. Here, we propose the use of a sub-word transfer function that converts disordered body-conducted speech into that of a healthy person. This transfer function was estimated using the cross-spectral method that makes use of each sub-word signal. Although speech from a disabled person was not available, speech sounds had previously been recorded, and our proposed system allows the recovery of these speech sounds. In the absence of any historical speech records, a transfer function is used to estimate the speech sounds from speakers such as a relative.

In applying the system, we investigated the following issues.

- Effectiveness of sub-word unit transfer functions made by cross spectrum method
- Examination for deciding sub-word unit

The system constructed for Japanese, so we examined several sub-word units.

- Phoneme
- Syllable and Semi-syllable
- Mora

Phonemes and semi-syllables are the smallest sub-word units. In pilot experiments, it was found that these do not estimate enough of each sub-word parameter of the cross spectrum transfer functions. Thus in further experiments, we examined the syllable and mora, which are

longer than the other candidates. These candidates were found to estimate stable parameters for each sub-word transfer function. Because the Japanese language is constructed of several moras, we chose the mora as the unit in our system. Next, we discuss what should be used in the system as the transfer function unit. In this paper, we discuss the recognition sub-word unit and making transfer functions for context independent models only. However, the system performance is expected to improve if transfer functions can be created for context dependent models, and recognition performance should improve accordingly.

3.4.3 Transfer function database

To construct a transfer function database, we need to consider the following issues.

- An estimate of how many transfer functions need each type of signal samples
- The problem of difference phonetic contexts for each sub-word environment

The cross spectrum method expects transfer function parameters to have only one set of signals for each pair of samples. However, these transfer functions have to use all contexts of the sub-word sequence when using an acoustic model for recognition and speech retrieval. To estimate a transfer function, we use all context samples to create a transfer function database. However, as samples often contain silence at the start and end of the sample, the transfer function is not able to capture the characteristics of the frequency magnitude. This problem is discussed in the next section. As the first step in the system, we focus on context-dependent sub-word transfer functions and creating transfer functions from one pair of set signals of speech and body-conducted speech for each sub-word. We have already explained that if a context dependent transfer function is used, the techniques used in the system are significantly improved.

3.5 Investigation of the effectiveness of transfer function with speech

In this section, we examine the effectiveness of a cross spectrum method in speech retrieval. If a recognition system contains recognition errors, it does not function correctly. To investigate this problem, we divided the experiment into two cases with different experimental conditions. One system carries out recognition correctly, while the other contains errors.

3.5.1 Experimental setup for speech retrieval experiments

Speech is recorded with a microphone placed 30 cm from the speaker. Body-conducted speech is extracted with an accelerator and its amplitude is then boosted by a suitable amplifier with the accelerator position set as the upper lip. These experiments focus only on the effectiveness of speech retrieval using the proposed method. This position is best for picking up body-conducted speech clearly with an accelerator. Each signal is recorded with 16 bit, 48 kHz sampling, and then both signals are synchronized after each signal is converted from 48 kHz to 16 kHz on a computer. In the experiment, words read by a 20-year-old male are recorded by the microphone.

One of the words is "Asahi (/a/, /sa/, /hi/)" and it is also contained in the JEIDA database with 100 locality names. This word has several different phonetics. The system uses Julian as the recognition decoder. The purpose of this experiment is to estimate only the boundary of each sub-word, because we use Julius for supervised recognition.

The recognition system consists of a 2-stage decoder with a decoding algorithm. The first stage uses a bi-phone and 2-gram model to calculate approximately the N-best results, while the second stage calculates details of each of the N-best results using a tri-phone and 3-gram model.

Recognition errors are generated from correct results, by changing correct to fail in each sub-word. The following labels are examined in this experiment.

- Correct: /a/, /sa/, /hi/
- Incorrect: /hi/, /hi/, /a/

These labels are used when esophageal speech is converted to retrieved speech.

3.5.2 Investigation of speech retrieval from body-conducted speech

Here we discuss details of the results of the retrieval experiment. Figure 12 shows the speech that is extracted using the microphone, while Figure 13 shows the body-conducted speech that is picked up with the accelerator. The upper parts of the figures show wave form data while the lower parts show the corresponding spectrograms. The speech is very clear, and thus speech characteristics such as formant frequency and high resolution frequency can be found. On the contrary, the body-conducted speech does not have these characteristics and this signal is not as clear as that of the normal speech. Comparing speech and body-conducted speech, the body-conducted signal cannot capture high frequency components of 2 kHz or more, which indicates that body-conducted signals do not have any formant frequency. Therefore, the body-conducted signal is not a naturally produced signal and is a lower quality signal compared with speech signals. Figure 14 shows the retrieved speech using correct recognition results, whereas Figure 15 shows the retrieved speech using incorrect recognition results. In Figure 14, we observe frequency retrieval at 2 kHz or more and formant frequencies. Focusing on each sub-word signal, each signal represents several formant frequencies using the sub-word unit transfer function. For this reason, it is clear that the system is effective. In Figure 15, we see that frequency retrieval at 2 kHz is not adequate to obtain the same retrieval results compared with Figure 14. However, each recognition result is not correct, and therefore, its signal contains other signal formant frequencies.

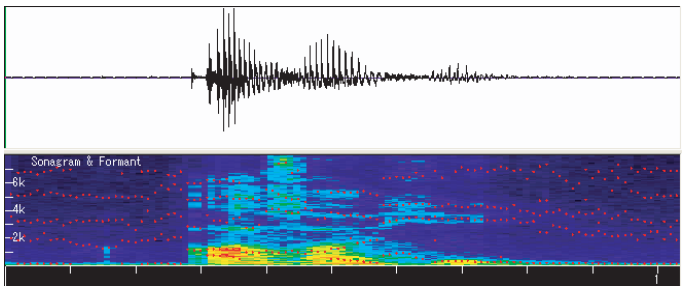


Fig. 12. Speech

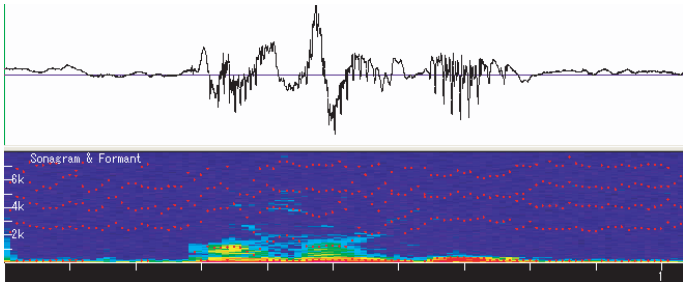


Fig. 13. Body-conducted speech

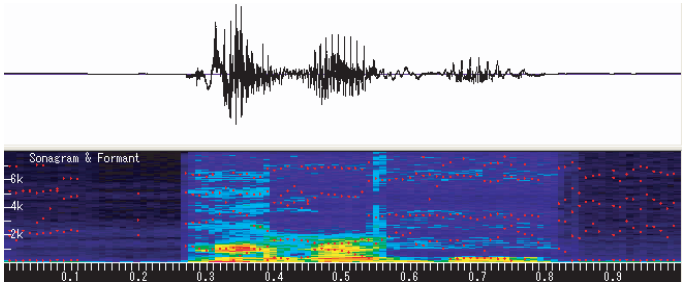


Fig. 14. Retrieved speech using correct recognition results

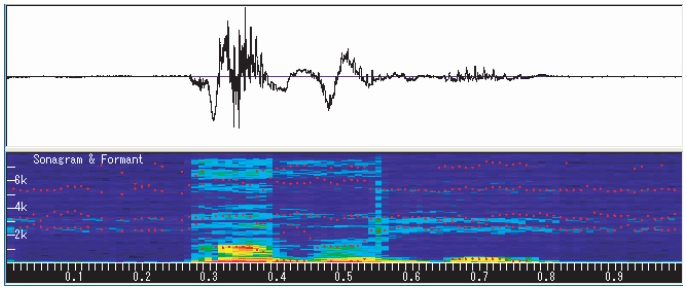


Fig. 15. Retrieved speech using incorrect recognition results with errors

4. Conclusion

First, we investigated a body-conducted speech recognition system for the establishment of a usable dialogue-type marine engine operation support system that is robust in noisy conditions, even in a low SNR environment such as an engine room. By bringing body-conducted speech close to audio quality, we were able to examine ways to raise the speech recognition rate. We introduced an adaptive processing method and confirmed the effectiveness of adaptive processing via small repetitions of utterances. In an environment of 98 dB SPL, improvements of 50% or above of recognition rates were successfully achieved within one utterance of the learning data and speech recognition rates of 95% or higher were attained. From these results, it was confirmed that this method will be effective for establishment of the present system.

Second, we have proposed a speech support system using body-conducted speech recognition. Such a recognition system can provide people with disorders related to cancer of the pharynx with a new speech communication tool for conversation. The system consists of a body-conducted speech recognition method and a transfer function database. The recognition system provides each sub-word and its duration per sentence in speech conversation. Based on this information, the system is able to retrieve the speech using the sub-word unit transfer function. In recognizing correct and erroneous results, we confirm each signal improvement based on its waveform and spectrogram. In particular, the experiments confirmed that retrieved speech of healthy people approximates the retrieval of speech signals with high frequency and formant information. In future work, we will apply the system to those with speech disorders, and the new system will examine the possibility

of a recognition system to assist disabled people with conversation and to estimate natural speech retrieval.

5. References

Matsushita, K. and Nagao, K. (2001). Support system using oral communication and simulator for marine engine operation., *Journal of Japan Institude of Marine Engineering*, Vol.36, No.6, pp.34-42, Tokyo.

Ishimitsu, S., Kitakaze, H., Tsuchibushi, Y., Takata, Y., Ishikawa, T., Saito Y., Yanagawa H. and Fukushima M. (2001). Study for constructing a recognition system using the bone conduction speech, *Proceedings of Autumn Meeting Acoustic Society of Japan* pp.203-204, Oita, October, 2001, Tokyo.

Haramoto, T. and Ishimitsu, S. (2001). Study for bone-conducted spcceh recognition system under noisy environment, *Proceedings of 31st graduated Student Mechanical Society of Japan*, pp.152, Okayama, March, 200, Hiroshima.

Saito, Y., Yanagawa, H., Ishimitsu, S., Kamura K. and Fukushima M.(2001), Improvement of the speech sound quality of the vibration pick up microphone for speech recognition under noisy environment, *Proceedings of Autumn Meeting Acoustic Society of Japan* I, pp.691 ~ 692, Oita, October, 2001, Tokyo.

Itabashi S. (1991), *Continuous speech corpus for research*, Japan Information Processing Development Center, Tokyo.

Ishimitsu, S., Nakayama M. and Murakami, Y.(2001), Study of Body-Conducted Speech Recognition for Support of Maritime Engine Operation, *Journal of Japan Institude of Marine Engineering*, Vol.39, No.4, pp.35-40, Tokyo.

Baum, L.E., Petrie, T., Soules, G. and Weiss, N. (1970), A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Annals of Mathematical Statistics*, Vol.41, No.1, pp.164-171, Oxford.

Ishimitsu, S. and Fujita, I. (1998), *Method of modifying feature parameter for speech recognition*, United States Patent 6,381,572, US.

Akimoto, H., Fujii, K., Mori H., and Kasuya H.(2002), Improvement of prosody and voice quality of esophageal speech, *in IEICE Technical Report*, SP2002-94, pp.59-64.

Nakamura, K., Toda, T., Saruwatari, H., and Shikano, K.(2007), A Speech Communication Aid System for Total Laryngectomees Using Voice Conversion of Body Transmitted Artificial Speech, *Journal of IEICE*, Vol.J90-D no.3, pp.780-787.

Ando, A., and Takagi, T.(2007), High-quality Speech Synthesis and Speech Processing Technology, *Journal of ICICE*, Vol.90, No.2, pp.91-94.

Fukushima, M., and Kido, K.(2007), Investigation of estimation error in impulse response by using cross spectral technique, *Journal of the ASJ*, Vol.55 N0.4, pp.265-274.

Morise, M., Irino, T., and Kawahara, H.(2007), Error Evaluation of Impulse Response Estimation by Cross Spectral Method Using Speech Signal, *Journal of IEICE*, Vol.J90-A N0.7, pp.559-566.

Kawahara, T., Lee, A., Kobayashi, T., Takeda, K., Minematsu, N., Itou, K., Ito, A., Yamamoto, M., Yamada, A., Utsuro, T., and Shikano K.(1999), Japanese Dictation Toolkit -1997 version-, *Journal of ASJ*, Vol.20 No.3, pp.233-239.

5

Modelling of Filled Pauses and Onomatopoeias for Spontaneous Speech Recognition

Andrej Žgank and Mirjam Sepesy Maučec
University of Maribor, Laboratory for Digital Signal Processing
Slovenia

1. Introduction

With the growing availability of various content provided over state-of-the-art digital media is speech recognition becoming one of the main core technologies (Billi et al., 1997; Žgank et al., 2002; Gupta et al., 2000; Sket et al., 2002). Its task is to minimize the needed effort to access the particular part of content. The main content categories can be grouped in the following way:

- broadcasted media,
- public and governmental content,
- entertainment,
- education,
- meetings,
- personal communication,
- personal repositories,…

The common point of all items is that characteristics of such spoken content widely diverge from type of speech, which is commonly found in spoken language resources used for training automatic speech recognition systems (Maddi et al., 2006; Marvi, 2006; Al-Haddad et al., 2006a; Al-Haddad et al., 2006b; Thangarajan et al., 2008). The main issue, which influences the quality of speech recognition, is the presence of spontaneous speech with all its special requests and characteristics. A speaker in such scenario can speak freely, without planning his/her speech. The vocabulary has size of several 10k words, which hardly depends on the properties of language involved. For less inflectionally and morphologically complex languages (e.g.: English, Spanish, Italian,…), the size of 64k vocabulary words can cover more than 99% of words in the test set (out-of-vocabulary (OOV) rate). On the other side are complex highly inflectional and agglutinative languages (e.g.: Finnish, Hungarian, Slovenian, Czech …), where the same size of vocabulary produces the OOV of 10% or even more.

In such cases present all various effects of spontaneous speech an additional parameter, which reduces the quality and performance of speech recognition for several percents. The applications where such problems can occur are: speech-to-speech translation system, "how can I help you?" telecommunication services, TV subtitling services, spoken content indexing services...

Real time TV subtitling service as one of the emerging services (Lambourne et al., 2004; Brousseau et al., 2003; Imai et al., 2000) in current and future society with increased proportion of elderly people is gathering on importance. The proportion of broadcasted content, which can't be immediately subtitled from content scripts, hardly depends on the

show's type. In a typical broadcast news show, approximately 50% to 75% of stories can be automatically subtitled using closed caption generated from the scripts. Example of such Slovenian evening news show script is given on Figure 1.

SLAVKO
PRIJETI PREPRODAJALCI OROŽJA, POKI V MESTU POLICIJSKI ZVOČNI EFEKTI
T- LJUBLJANA, ATENE
X- Olimpijske igre
EDITA
ZA PROMOCIJO SLOVENIJE V ATENAH DESETKRAT MANJ DENARJA KOT V SYDNEYJU
02, Nap 0(Sprejem Goričanov K2, (Čurlič, B 3 ____ NL, 0:23, 6:59:50 38" Stat_Aired ____SPREMENIL: haskaj _____KDAJ: 08/05/04 18:37:57 T- EDITA M. CETINSKI
X- SLAVKO BOBOVNIK
T- NOVA GORICA /zg.
K2-DVOPLAN (SLAVKO) Dober dan, cenjene gledalke in spoštovani gledalci.
K2-DVOPLAN (EDITA) Lepo pozdravljeni.
SLAVKO
Upajmo, da se bomo tako, kot so se danes veselili Novogoričani, BETA veselili tudi mi, ko se bodo naša dekleta in fantje vračali iz Aten. Na sprejemu slovenskih nogometnih prvakov, ki so včeraj s kar pet proti nič premagali danske prvake, se je danes zagotovo zbralo kakih 1000 ljudi.

03, izjava župana _____ TONSKO, B 4 ____ *, 0:09, 7:00:13 ___Stat_Aired ____SPREMENIL: golob _____KDAJ: 08/05/04 18:37:38 T- MIRKO BRULC
X- župan Mestne občine Nova Gorica
KONČA: ... Dragi Novogoričani. 30. aprila je "Evropa gledala" v Novo Gorico.
Danes pa po vaši zaslugi zopet gleda v Novo Gorico.
04, Nap 1 _____ K2, ____ _ ____ NL, 0:23, 7:00:22 ___Stat_Aired ____SPREMENIL: nakrst _____KDAJ: 08/05/04 18:42:28 K2, DVOPLAN
SLAVKO
O podrobnostih v športu, saj so prve minute Dnevnika namenjene precej manj prijetnim dogodkom.
EDITA
Na Slovenskem zunanjem ministrstvu so povedali, da so po njihovem mnenju navedbe v hrvaški diplomatski noti napačne in da so bili naši policisti ob nedavnih incidentih v Piranskem zalivu v vodah pod slovenskim nadzorom.
SLAVKO
Odgovora na noto pa na ministrstvu še niso napisali.

Fig. 1. Evening TV news show script, a part of the Slovenian BNSI Broadcast News database.

The remaining part of the show isn't covered, as it contains live conversations (e.g. interviews, talk shows), where closed captions can't be generated from scripts or scenarios. Example of such script part is shown on Figure 1, denoted as section "03 izjava župana", where only the last few seconds are transcribed as guideline for the director. These parts of shows must be covered with dedicated methods as is spontaneous speech recognition. Two methods can be used for producing closed captions: respeaking, where a highly trained operator respeakes all utterances in an of-the-shelf dictation system or a fully automated subtitling system, which must process the entirely show, usually in several steps.

Automatic recognition of spontaneous conversations is a very challenging task. There are three major groups of disfluencies in spontaneous speech that influence the quality and performance of any spontaneous speech recognition system:

- Filled pauses (FP): short words, which appear as interjection – e.g.: uh, aaa. They are language dependent.
- Word repetitions: disfluencies used by the speaker to gain time before continuing with the sentence.
- Sentence restarts: speaker pronounces the initial part of a sentence and then starts over again with a new initial part.

Figure 2 shows example of spontaneous sentence ("mirna sobota ki ee so jo mnogi") from Slovenian BNSI Broadcast News database. The shown sentence encompasses one filled pause – "eee". The ratio of disfluencies in spontaneous speech hardly depends on the situation. In case when the speech is prepared in advance, there are far less disfluencies than in case when spontaneous speech is used in everyday situation.

The presented characteristics of spontaneous speech influence both types of models in a system – acoustic and language model. On the other side, the accents mainly influence the performance of acoustic models. Spontaneous conversation can involve a high degree of accented speech, depending of the discourse properties. In case of broadcast news language resources various groups of interviews include such discourse.

Fig. 2. Spontaneous sentence from Slovenian BNSI Broadcast News speech database.

Spontaneous speech is also a challenging task for language modelling. It is characterized by unconstrained speaking style, frequent grammatical errors, hesitations, starts-over, etc. Another problem is a limited amount of training data. The main source is audio transcription. Unfortunately, sources of written data do not exhibit characteristics of spoken language.

The research work presented in this chapter is oriented on modelling of filled pauses and onomatopoeias for spontaneous speech recognition system. A previously proposed filled pauses acoustic modelling approach will be further improved with an advanced training procedure. In addition to normally accented speech, also a heavily accented spontaneous speech of a non-native speaker will be included in the experiment. Filled pauses are one of the most frequent categories of spontaneous speech effects, which are present in real-life spoken language resources and will be as such included in our experiments. Onomatopoeias as another category are less frequent, but still very challenging for modelling. We have grouped both categories in one, called filled pauses. Although filled pauses and onomatopoeias don't carry any true semantic information, it is still necessary to include them in modelling for speech recognition. Each filled pause disrupts the sequence of words, which is estimated with the acoustic and language model and so influences the overall accuracy of speech recognition system. In addition, disfluencies in spontaneous speech are often indicators of turn taking in a dialog, and can be as such used for dialog management in voice driven telecommunication services. The methods proposed for modelling of filled pauses will be also evaluated on heavy accented speech, to show that modelling of filled pauses plays even more important role in such case of conversation.

The level of accented speech usually depends on the speaker and its role in the discourse. In addition to these properties, the language also plays an important role. There are some languages, where a large number of various accents can be found. Slovenian is one of such languages, with approximately 50 different accents. This makes any accent modelling an additionally challenging task.

The chapter is organized as follows: the current state-of-the-art is described in Section 2. Various filled pauses modelling approaches are presented in Section 3. The native and non-native spoken language resources are introduced in Section 4. The experimental design used for evaluation is described in Section 5. Section 6 contains the results of the speech recognition experiments, while the conclusion and directives for future work are given in Section 7.

2. Overview of current research work on topic of spontaneous speech recognition

In the last few years is the research area of spontaneous speech recognition gathering on importance. One of the prerequisites for this development was the increase in CPU power, as are the algorithms for spontaneous speech recognition very demanding on processing power.

In the area of acoustic modelling of filled pauses, several authors presented successful approaches, how to address this topic. The first group of methods is based on Gaussian Mixture Modelling (GMM) (Wu & Yan, 2004; Wu & Yan, 2001; Rangarajan & Narayanan, 2006). There are two main approaches possible. In the first approach, for each type of filled pauses a separate GMM model is build. The number of mixtures depends on the availability of spoken material per class. A separate class is used for modelling of normal spontaneous speech without any filled pauses. As the end results a system with multi GMM is being used for explicit (see Section 3) recognition of filled pauses in spontaneous speech. The second approach is based on only two GMM models. The first one represents filled pauses and the second one normal spontaneous speech. The main advantage of the second approach is that it is simpler to collect adequate amount of training material per class to train the GMM models. It also reduces the classification error between various types of onomatopoeias, as it can be sometimes extremely difficult to label separate sounds correctly. In general, the second approach yields better speech recognition results due to its higher modelling capability.

The second major group is based on modelling with Hidden Markov Models (HMM) (Furui et al., 2005; Stouten et al., 2006; Seiichi & Satoshi, 2007), usually in an implicit way (see Section 3 for details). The performance of this group of approaches depends on the quality of transcriptions of spoken language resources. Each filled pause must be correctly labelled and transcribed to be able to model it with an HMM model. There are several methods possible how can a filled pause be represented with an HMM. One approach is to use separate HMM models for filled pauses. Another approach uses the same HMM acoustic models for filled pauses and spontaneous speech. The second approach is more difficult and complex as acoustic-phonetic properties of both types usually differ. Therefore complex modelling approaches are needed to reduce this discrepancy. It is also possible to combine the above presented methods in one system.

The specifics of spontaneous speech presented above for acoustic modelling are also reflected on language modelling. Disfluencies (repetitions, hesitations, and sentence restarts) distinguish spontaneous from read speech to a great extant. N-grams base their word prediction on a local context of N-1 previous words. Early psycholinguistic experiments found that human subject asked to guess next word in the transcription a spontaneous speech required more guesses for words that had been proceeded by a hesitation (Goldman, 1968). The experiment indicates the difficulties of transition from modelling read speech to modelling spontaneous speech.

Disfluencies corrupt this context. First, the idea was to remove disfluencies from the context. Based on experiments it has been shown that simple clean-up is not the right way to recover the fluent order of meaningful words (Duchateau et al., 2004). If we eliminate disfluencies completely, we would lose some information.

In (Duchateau et al., 2004) the authors allow the system to pick the most probable option when both a context with and without disfluencies are available. In case of repetitions the results were improved significantly by offering the system the choice between removing or not removing the disfluency from the prediction context. For hesitations and restarts this method results in a small deterioration of the recognition rate. The research was later extended by developing a specialized preprocessor which operated independently of the search and which searches for filled pauses on the basis of acoustic and prosodic features that are not accessible to the recognizer (Stouten et al., 2006). A filled pauses detector was built. Two strategies for incorporating the posterior probabilities at the output of this detector into the search engine were proposed.

Filled pauses and onomatopoeias don't carry any true semantic information, but should be incorporated into the language model. The biggest difficulty is that statistical language models typically have very limited context, and by keeping filled pauses and onomatopoeias in context, information bearing word is lost. In (Stolcke et al., 1999) they are demarcated by events surrounding the words. They refer to them as Hidden Word-level Events (HWE). Models of HWE capture the specific prosodic characteristics of HWEs, such as intonation and duration patterns. The information from prosodic features was combined with statistical language models that describe the distribution of HWE in relation to words, part-of-speech, and other syntactical and lexical units.
Adaptation to speaker-dependent disfluencies was studied to adopt a system for disfluency removal. Disfluency removal makes sentences shorter, less ill-formed and thus facilitates the downstream processing by natural language understanding components such as machine translation or summarization (Honal & Schultz, 2005). The probability that a word is disfluent is composed of a weighted sum over the six models. The most prominent were the model of the length of the deletion region of a disfluency and the model of the position of a disfluency. Gradient descent method was used to automatically optimize the parameter weights.
Speaker-produced disfluencies were identified in a conditional random field-based approach (Fitzgerald et al., 2009). The authors emphasize false start regions, which are often missed in current disfluency identification approaches as they lack lexical or structural similarity to the speech immediately following. They find that combining lexical, syntactical, and language model-related features with the output of the state-of-the-art disfluency identification system improves overall word-level identification of these and other errors.
Although there has been significant work devoted to some spontaneous speech phenomena, we are still looking for an accurate and efficient language models for speech disfluencies.

3. Spontaneous speech and modelling of filled pauses and onomatopoeias

There are two different types of filled pauses acoustic modeling from the speech recognizer's point of view. In the first case filled pauses are detected using an external module (e.g. GMM classification (Wu &Yan, 2004)), and speech recognizer than process only the part of speech without filled pauses (Figure 3).

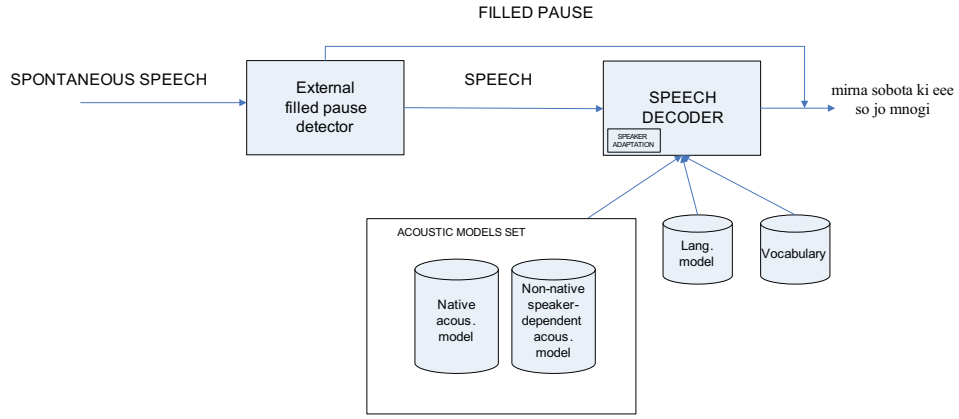


Fig. 3. Explicit modelling of filled pauses in a speech recognition system.

In the second case are acoustic models for filled pauses part of the main speech recognition decoding process. This is called implicit modelling of filled pauses (Figure 4).

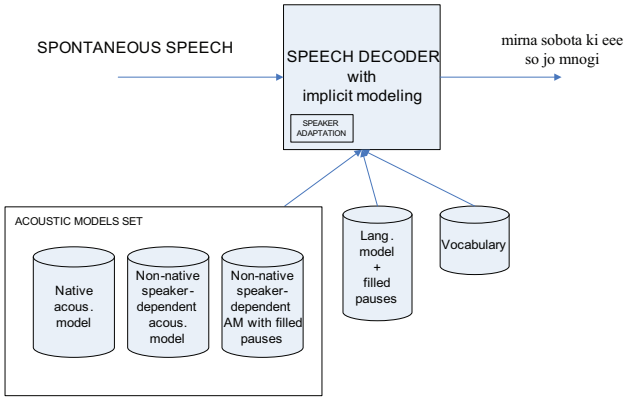


Fig. 4. Implicit modelling of filled pauses in a speech recognition system.

3.1 Implicit modelling of filled pauses

In the basic acoustic modelling approach (AM1), all filled pauses use only one acoustic model. This results in combining all filled pauses, regarding their acoustic-phonetic properties, into one common model. In such a way, acoustic training material is grouped together, which is important in case of infrequent filled pauses (see Table 4). The drawback is that the modelling of acoustic diversities isn't taken into account. In our case, where the acoustic modelling was performed using the HMM, one three state left-right model was applied. The acoustic model for filled pauses was used as context-independent one and was as such also excluded from the phonetic decision tree based clustering of triphone acoustic models (see Section 5 for more details).

The second implicit acoustic modelling approach (AM2) uses a separate acoustic model for each type of filled pauses. Advantage is that such model covers all acoustic-phonetic properties of one type of filled pauses, but the problem can be with the amount of training material available for infrequent types of filled pauses. As for the first example, the HMM models are context independent.

The third kind of implicit modelling (AM3) is based on general acoustic models that are also used for speech modelling. Each filled pause is modelled with the speech acoustic models, according to its acoustic-phonetic properties. This solution usually assures enough training material for all types of filled pauses. The disadvantage lies in the fact that acoustic-phonetic properties of speech differ from those of filled pauses. The main difference is caused by duration of phonemes and levels of pitch. In case of this modelling approach, some of HMM models are context-dependent and therefore included in phonetic decision tree based clustering. The examples of all three implicit modelling approaches are presented in Table 1.

There are three different filled pauses present in Table 1: eee, eem, and mhm. In case of AM1 acoustic models all three filled pauses are modelled with the common context-independent acoustic model "filler". When AM2 acoustic models are applied, each filled pause has its own context-independent acoustic model for filled pauses (e.g. filled pause eee is modelled with "eee" acoustic model). In the last case, when AM3 acoustic models are applied each

filled pause is modelled with context-dependent acoustic models for regular words – filled pause mhm is modelled with acoustic models "m h m" for regular words.

Filled pause	AM1	AM2	AM3
Eee	Filler	eee	e e
Eem	Filler	eem	e m
Mhm	Filler	mhm	m h m

Table 1. Three different approaches of implicit acoustic modelling of filled pauses.

3.2 Implicit modelling of filled pauses based on phonetic broad classes

Considering all presented properties of described acoustic modelling approaches, a new method (AM4) how to model filled pauses was proposed in (Žgank et al., 2008). The basic idea is to use phonetic broad classes to model filled pauses. Phonetic broad classes are defined for each specific language, either by an expert phonetician or in a data-driven way. Phonemes with similar properties (e.g. open vowels) are grouped together in a particular phonetic broad class.

Class-01 *i i:*
Class-02 *m n v l b*
Class-03 *E i O u: E: e: ehr*
Class-04 *i: e:*
Class-05 *O u: o: W o w d-n ehr O:*
...

Fig. 5. Slovenian phonetic broad class, defined in a data-driven way.

Example of Slovenian phonetic broad classes, defined in a data-driven way (Žgank et al., 2005a; Žgank et al., 2003) is shown on Figure 5. One of the smallest phonetic broad classes is Class-01 with only two members "i" and "i:". On the opposite side are phonetic broad classes, which have several members, as for example Class-05 with 9 members.

Instead of using a separate acoustic model as in case of AM2, a group of acoustic models is used to model filled pauses. Groups should be defined in a way that they incorporate acoustically similar filled pauses with enough training material. The analysis of the training set showed (see Table 4) that 4 different categories should be defined: vowels, voiced consonants, unvoiced consonants, and mixed group. The last one is used for those filled pauses that can't be reliably categorized into the first three groups. The advantage of this method is in the fact that are the acoustic models of filled pauses still separated from the acoustic models of speech. Therefore, they can better model peculiarities of filled pauses that strongly differ from speech. An example, how filled pauses are modelled with the AM4 method is shown in Table 2.

Filled pause	AM4
Eee	Vowels
Eem	Mixed
Mmm	voiced consonants
Sss	unvoiced consonants

Table 2. Modelling of filled pauses using the method based on phonetic broad classes.

In AM4 approach, each filled pause belongs to one of the possible phonetic broad class categories (vowels, mixed, voiced consonants, unvoiced consonants). The filled pause eem, which pronunciation is combination from vowels ("e") and consonants ("m") is member of category mixed. On the other side, the pronunciation of filled pause eee contains only vowels; therefore it is a member of the first category vowels. The AM4 method already proved promising results. The current focus is to evaluate the method with improved training procedure and on heavy accented speech.

4. Slovenian BNSI Broadcast News speech and text corpora

The primary language resource used during these experiments was the Slovenian BNSI Broadcast News database (Žgank et al., 2005b). The BNSI database was designed in cooperation between University of Maribor, Slovenia and the Slovenian national broadcaster RTV Slovenia. The raw audio material was acquired from the archive of the broadcast company on DAT and DVD-R media. The captured audio signal was manually segmented, annotated and transcribed with tool Transcriber (Barras et al., 2001), according to recommendations on building Broadcast News spoken language resources.
The speech corpus comprehends two different types of TV-news shows. The first type is evening news where general overview of daily events is given. The second types of show are late night news where major events of the day are analyzed. In this type of news show are frequent longer interviews (up to 10 minutes), with high proportion of spontaneous speech.
The speech corpus consists of 42 news shows, which account for 36 hours of speech material. This material is further grouped into three sets: training, development and evaluation, respectively. The size of the training set is 30 hours, whereas the size of the development and evaluation set is 3 hours each. Altogether 1565 different speakers are present in the BNSI database. The majority, 1069 of them, are male, while 477 are female. The gender of remaining 19 speakers was annotated as unknown. With usage of additional preprocessing steps on level of manual transcriptions the amount of training material which was prior excluded from the training set was reduced. Detailed analysis of speech recognition results showed statistically significant improved performance due to this additional step.
It addition to the speech corpora, the text corpora (scenarios, transcriptions of speech corpus) was built. The text corpus is needed for developing the baseline set of language models. The Slovenian Vecer Newspaper text corpus was additionally incorporated in the language modelling. Properties of the BNSI Broadcast News database are given in Table 3.

speech corpus:	
total length(h)	36
number of speakers	1565
number of words	268k
test corpus:	
number of words	11M
distinct words	175k

Table 3. Slovenian BNSI Broadcast News speech and text database.

The evaluation set of the BNSI Broadcast News speech database is composed from 4 broadcasts in total length of approx. 3 hours. Typical broadcast news show comprises

various types of speech: read or spontaneous, in studio or over telephone environment, with or without background (Žgank et al., 2005b; Schwartz et al., 1997) (Figure 6).

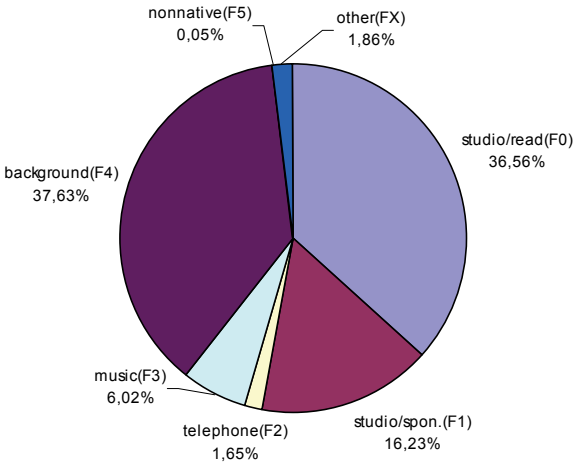


Fig. 6. Ratio of various focus conditions in the BNSI speech database.

The goal in this experiment was to efficiently evaluate the acoustic modelling of filled pauses. Therefore only the utterances with spontaneous speech in clean studio environment (F1-focus condition (Schwartz et al., 1997)) were included in the evaluation set. There were 343 utterances with 3287 words in the evaluation set. The analysis showed that there were 155 different filled pauses in this evaluation set, which represent 4.72% of it. The training set comprises 24 broadcasts.

An analysis of all filled pauses that were found in the training set was also carried out. Those filled pauses with frequency higher than 5 are presented in Table 4.

Filled pause	Frequency
eee	1833
Sss	60
Mmm	43
Eem	40
Zzz	21
Uuu	16
Ooo	14
Vvv	12
Ttt	12
Aaa	12
Nnn	10
Iii	9
Ppp	8
Mhm	7
Eeh	7

Table 4. Statistics of filled pauses in the training set.

The most frequent filled pause in the training corpus is "eee", with frequency 1833. The other filled pauses are far less frequent. The second one in Table 4 has frequency 60. There are altogether 15 filled pauses, which frequency is higher than 5. This distribution of frequencies between filled pauses support the idea of joining phonetically similar filled pauses in a same acoustic model, as the lack of appropriate training material for modelling of filled pauses can be foreseen.
The secondary spoken language resource was used for modelling and evaluating heavy accented speech. For this experiment, the Slovenian SINOD speech database (Žgank et al., 2006a) was used. The SINOD database was developed as a supplement to the BNSI Broadcast News database. It consists of two TV interviews, the first one with Russian non-native speaker (Table 5) and the other one with English non-native speaker of Slovenian. The same structure and transcription rules were applied as in the BNSI database. Here, only the part with the Russian non-native speaker of Slovenian was involved in the training and evaluation procedure. The secondary spoken language resource plays an important resource as it involves a high proportion of accented filled pauses, due to the non-native speaker involved. The presented spoken language resource has the drawback that only one speaker and its speaking style is involved in the heavy accented speech experiments. But the fact is that such spoken language material is extremely difficult to collect, especially for languages with smaller number of speakers. To reduce this characteristic of non-native spoken language resource, adaptation procedures presented in Section 5 were additionally incorporated.

speech corpus:	SINOD
total length(mm:ss)	28:20
number of sentences	642
distinct words	1010
test corpus:	
test set length (mm:ss)	8:36

Table 5. Slovenian non-native database SINOD (Russian speaker).

5. Experimental design

The experimental design (Figure 7) is based on continuous density Hidden Markov Models for acoustic modelling and on n-gram statistical language models.

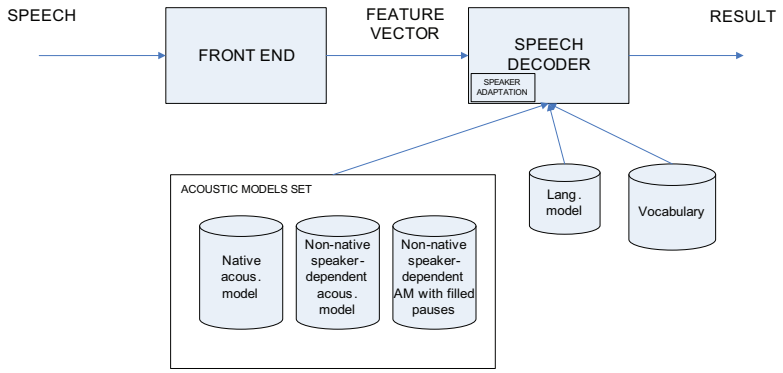


Fig. 7. Block diagram of experimental speech recognition system.

The core module is a speech decoder, which needs three data sources for its operation: acoustic models, language model and lexicon.

5.1 Acoustic modelling

The frontend was based on mel-cepstral coefficients and energy (12 MFCC + 1 E, delta, delta-delta). The size of feature vector was 39. Also, the cepstral mean normalization was added to the feature extraction to improve the quality of speech recognition. The manually segmented speech material was used for training and speech recognition. This was necessary to exclude any influence of errors that could occur during an automatic segmentation procedure. The developed baseline acoustic models were gender independent. The training of baseline acoustic models was performed using the BNSI Broadcast News speech database. The procedure was based on common solutions (Žgank et al., 2006b). First the context independent acoustic models with mixture of Gaussian probability density function (PDF) were trained and used for force alignment of transcription files. In the second step, the context independent acoustic models were developed once again from scratch, using the refined transcriptions. The context-dependent acoustic models (triphones) were generated next. The number of free parameters in the triphone acoustic models, which should be estimated during training, was controlled with the phonetic decision tree based clustering. The decision trees were grown from the Slovenian phonetic broad classes that were generated using the data-driven approach based on phoneme confusion matrix (Žgank et al., 2005a; Žgank et al., 2003). Three final sets of baseline triphone acoustic models with 4, 8 and 16 mixture Gaussian PDF per state were generated. As some additional training data was won from the pool of outliers in comparison with the system described in (Žgank et al., 2008), additional training iterations were applied to context-dependent acoustic models. These transcriptions preprocessing steps showed significant improvement of log-likelihood rate per acoustic model according to an analysis.

Our main task was the acoustic modelling of filled pauses. To exclude from the experiments influence of inter-speaker variations in pronouncing filled pauses, only the speaker independent acoustic models were applied for native test set.

For the heavy accented speech with the non-native set using the SINOD speech database, the baseline BNSI acoustic models were first adapted to particular speaker using the Maximum Likelihood Linear Regression (MLLR) (Leggetter & Woodland, 1995) procedure. The MLLR was used in an iteratively way. During the first iteration, acoustic models were adapted on general transcription. Thereafter the forced realigning procedure was used to improve the general transcriptions for a particular speaker. During the second iteration, the improved transcriptions were used for MLLR speaker adaptation. In the last step of modelling heavy accented speech, all approaches for modelling filled pauses were applied to the set of speaker dependent acoustic models.

The standard one-pass Viterbi decoder with pruning and limited number of active models was used for speech recognition experiments in the next section. We applied additional fine tuning of decoder parameters on combined development set in comparison to the system described in (Žgank et al., 2008), to further improve the performance of speech recognition system.

5.2 Language modelling and vocabulary

Language models were built using corpora of written language and transcribed speech. For LM training three different types of textual data were used: Vecer (corpus of newspaper

articles in period 2000-2002), iNews (TV show scripts in period 1998-2004) and BN-train (transcribed BNSI acoustic training set). The interpolation coefficients were estimated based on EM algorithm using a development set. The language model is based on bigrams. The vocabulary contained the 64K most frequent words in all three corpora. The lowest count of a vocabulary word was 36. The out-of-vocabulary rate on the evaluation set was 4.22%, which is significantly lower than for some other speech recognition systems built for highly inflectional Slovenian language (Žgank et al., 2001; Rotovnik et al., 2007). A possible reason for this is the usage of text corpora with speech transcriptions for language modelling.
Two types of language models were built. In the first model (LM1), all filled pauses and onomatopoeic words were mapped into unique symbol, which was considered as non-event, and can only occur in the context of a bigram and was given zero probability mass in model estimation. In the second model (LM2) filled pauses and onomatopoeic words were modelled as regular words.

	LM1	LM2
λ(BN-train)	0.2619	0.2665
λ(INews)	0.2921	0.2941
λ(Vecer)	0.4459	0.4392
perplexity	410	414

Table 6. Statistics of language models used for modelling filled pauses.

Language models built on the Vecer newspaper text corpus has the highest interpolation weight (0.4459 and 0.4392) for both types of language models. The interpolation weights for two other language models (iNews and BN-train) are similar. The perplexities of language models, calculated on the evaluation set were 410 and 414, respectively. The higher value for LM2 is due to the unmapped filled pauses.

6. Results

The proposed method of acoustic modelling of filled pauses will be evaluated indirectly with word accuracy, using the speech recognition results. These speech recognition results will be also used to compare the modelling methods for normal and heavy accented speech. The word accuracy is defined as:

$$Acc(\%) = \frac{H - I - D}{N} \cdot 100 \tag{1}$$

where H denotes the number of correctly recognized words, I the number of inserted words, D the number of deleted words, and N the number of all words in the evaluation set. First, three different versions of the baseline system without modelling of filled pauses were evaluated on normal speech, to check which system's topology performs best (Table 7).

	Acc(%)
Baseline 4 PDF	50.90
Baseline 8 PDF	55.82
Baseline 16 PDF	62.15

Table 7. Speech recognition results without modelling of filled pauses for three different topologies of acoustic models recognizing normal speech.

The simplest topology of acoustic models with 4 Gaussian PDF mixtures per state performed worst, with the 50.90% accuracy. When the number of mixtures was increased to 8 per state, the accuracy improved to 55.82%. The last baseline speech recognition configuration with 16 Gaussian mixtures achieved the best result with word accuracy of 62.15%. Thus the speech recognition performance was increased for 11.25% absolute. The relatively low performance of all three baseline systems is mainly due to the following facts: highly inflectional Slovenian language with high out-of-vocabulary rate, completely spontaneous type of conversations in the evaluation set and limitations of using speaker-independent acoustic models for this very complex speech recognition task. The disadvantage of the topology with 16 Gaussian mixtures per state, which yield the best result, is its complexity with high number of free parameters, which must be estimated. This results in increased computation time. The increased complexity of training procedure, presented in Section 5, improved the performance for approximately 5% in overall if compared to system applied in (Žgank et al., 2008).

In the next step of evaluation four different filled pauses modelling techniques (AM1-AM4) were tested. Appropriate language models (LM1, LM2) were used in combination with the correct type of acoustic models. The results are presented in Table 8.

	Acc(%)
AM1+LM1	62.73
AM2+LM1	62.96
AM2+LM2	62.98
AM3+LM1	63.60
AM3+LM2	64.37
AM4+LM1	64.95

Table 8. Speech recognition results without and with acoustic modelling of filled pauses.

Small improvement of recognition performance was already denoted for basic modelling of filled pauses on normal speech. The combination of AM1 and LM1 models increased the accuracy to 62.73%. Similar improvement of accuracy was achieved with the AM2 acoustic models, when LM1 and LM2 language models were used – the accuracy was 62.96% and 62.98% respectively. There was almost no influence of the language model type on the normally accented speech recognition performance. In case of AM3 acoustic models were filled pauses modelled in combined mode with normal speech. The evaluation of this approach showed word accuracy of 63.60% and 64.37% for each particular language model LM1 and LM2. In this case, the version of language model played an important role.

The last evaluation step for normally accented speech was focused on AM4 acoustic models where the filled pauses were modelled with phonetic broad classes according to their acoustic-phonetic properties. This approach achieved the best overall result with word accuracy of 64.95%. The baseline system performance was improved for 2.80% absolutely. Due to the improved training procedure, the improvement was smaller as in case of system described in (Žgank et al., 2008), although it was still statistically significant.

In the last step of evaluation, the heavy accented speech originating from the SINOD database was tested. The results for this case are presented in Table 9.

In case of SINOD database only the AM4 approach of modelling filled pauses was tested, as it already proved to be the most efficient one. The baseline SINOD system achieved the word accuracy of 65.74%. The improvement in comparison to the baseline system is result of

	Acc(%)
SINOD baseline	65.74
SINOD AM4	67.31

Table 9. Speech recognition results for heavy accented speech without and with filled pauses modelling.

applying MLLR procedure, although the increase of word accuracy is smaller than usual for speaker adaptation. The possible cause for this is the non-native origin of test speaker. In case, when the filled pauses were modelled using the proposed phonetic broad classes approach, the word accuracy increased to 67.31%. Thus the overall improvement for heavy accented non-native speech was 1.57%. The improvement is smaller as in case of native speech, but it still show, how important it is to model the filled pauses.

7. Conclusion

The new speech recognition system achieved statistically significant improvement of word accuracy in comparison with the previous version. The obtained speech recognition results clearly showed how important it is to adequately model filled pauses and onomatopoeias in spontaneous speech on level of acoustic and language models. The detailed analysis of speech recognition performance on filled pauses in non-native speech showed that there is still some room for improvements due to the complexity of this task.
The future work will be focused on various data-driven approaches, which will take into account the difference in pronouncing filled pauses and onomatopoeias in native and non-native speech. The detailed analysis of speech recognition results namely showed that this could further improve the performance of our system.

8. Acknowledgements

The work was partially funded by Slovenian Research Agency, under contract number P2-0069, Research Programme "Advanced methods of interaction in telecommunication".

9. References

Al-Haddad, S. A. R., Salina Abdul Samad, Aini Hussein, (2006). "Automatic Segmentation and Labeling for Continuous Number Recognition". *WSEAS TRANSACTIONS on SIGNAL PROCESSING*, Issue 9, Volume 2, September 2006.

Al-Haddad, S. A. R., Salina Abdul Samad, Aini Hussein, M. K. A. Abdullah, (2006). "Automatic Segmentation and Labeling for Malay Speech Recognition". *WSEAS TRANSACTIONS on SIGNAL PROCESSING*, Issue 9, Volume 2, September 2006.

Barras, C., E. Geoffrois, Z. Wu, and M. Liberman, (2001). "Transcriber: Development and use of a tool for assisting speech corpora production". *Speech Communication*,Vol. 33, Issues 1-2, 5-22.

Billi, R., Castagneri, G., Danieli, M., (1997). Field trial evaluations of two different information inquiry systems. *Speech Communication*, Volume 23, Issues 1-2, October 1997, Pages 83-93.

Brousseau, J., Beaumont, J.F., Boulianne, G., Cardinal, P., Chapdelaine, C., Comeau, M., Osterrath, F., Ouellet, P., (2003). "Automatic Closed-Caption of Live TV Broadcast News in French", *Proc. Eurospeech 2003*, Geneva, Switzerland.

Duchateau, J., T. Laureys, P. Wambacq, (2004). Adding Robustness to Language Models for Spontaneous Speech Recognition, *In Proc. ISCA Workshop on Robustness Issues in Conversational Interaction*, Norwich, UK.

Fitzgerald, E., K. Hall, F. Jelinek, (2009). Reconstructing False Start Errors In Spontaneous Speech Text. *In Proc. of the 12th Conference of the European Chapter of the ACL*, pp.255–263, Athens, Greece.

Furui, S., M. Nakamura, T. Ichiba and K. Iwano, (2005). "Analysis and recognition of spontaneous speech using Corpus of Spontaneous Japanese" *Speech Communication*, vol.47, pp.208-219.

Goldman-Eisler, F., (1968). *Psycholinguistics: Experiments in Spontaneous Speech*, New York: Academic Press.

Gupta, V., Robillard, S., Pelletier, C., (2000). Automation of locality recognition in ADAS plus, *Speech Communication*, Volume 31, Issue 4, August 2000, Pages 321-328.

Honal, M., T. Schultz, (2005). Automatic Disfluency Removal On Recognized Spontaneous Speech - Rapid Adaptation To Speaker-Dependent Disfluencies. *Proc. of ICASSP, 2005*, vol 1, pp. 969-972.

Imai, T., Kobayashi, A., Sato, S., Tanaka, H., and Ando, A., (2000). "Progressive 2-pass decoder for real-time broadcast news captioning", *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp.1937-1940, Istanbul, Turkey.

Lambourne, A., J. Hewitt, C. Lyon, S. Warren, (2004). "Speech-Based Real-Time Subtitling Services", *International Journal of Speech Technology*, Vol. 7, Issue 4, pp. 269-279.

Leggetter, Woodland, (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Comput. Speech Lang*. v9 i2. 171-185.

Maddi, A., A. Guessoum, D. Berkani, (2006). "Noisy Speech Modelling Using Recursive Extended Least Squares Method". *WSEAS TRANSACTIONS on SIGNAL PROCESSING*, Issue 9, Volume 2, September 2006.

Marvi, H., (2006). "Speech Recognition Through Discriminative Feature Extraction". *WSEAS TRANSACTIONS on SIGNAL PROCESSING*, Issue 10, Volume 2, October 2006.

Rangarajan, V., S. Narayanan, (2006). "Analysis of disfluent repetitions in spontaneous speech recognition", *Proc. EUSIPCO 2006*, Florence, Italy.

Rotovnik, T., Sepesy Maučec, M., Kačič, Z, (2007). "Large vocabulary continuous speech recognition of an inflected language using stems and endings". *Speech communication*, 2007, vol. 49, iss. 6, pp. 437-452.

Schwartz, R., H. Jin, F. Kubala, and S. Matsoukas, (1997). "Modeling those F-Conditions - or not", in *Proc. DARPA Speech Recognition Workshop 1997*, pp 115-119, Chantilly, VA.

Seiichi, N., K. Satoshi, (2007). "Phenomena and acoustic variation on interjections, pauses and repairs in spontaneous speech". *The Journal of the Acoustical Society of Japan*, Vol.51, No.3, pp. 202-210.

Sket, G., B. Imperl (2002). M-vstopnica – uporaba avtomatskega razpoznavanja govora v praksi. *Jezikovne tehnologije 2002*, Inštitut Jožef Stefan, Ljubljana.

Stolcke, A., E. Shriberg, D. Hakkani- Tür, G. Tür, (1999). Modeling The Prosody Of Hidden Events For Improved word Recognition, *In Proc. EUROSPEECH*, vol. 1, pp. 307–310, Budapest.

Stouten, F., J. Duchateau, J.P. Martens, P. Wambacq, (2006). "Coping with disfluencies in spontaneous speech recognition: Acoustic detection and linguistic context manipulation". *Speech Communication* 48(11): 1590-1606.

Thangarajan, R., A.M. Natarajan, M. Selvam, (2008). "Word and Triphone Based Approaches in Continuous Speech Recognition for Tamil Language". *WSEAS TRANSACTIONS on SIGNAL PROCESSING*, Issue 3, Volume 4, March 2008.

Wu, Chung-Hsien, Yan, Gwo-Lang, (2001). "Discriminative disfluency modeling for spontaneous speech recognition", *In: EUROSPEECH-2001*, Aalborg, Denmark, pp. 1955-1958.

Wu, C. and Yan, G. (2004). "Acoustic Feature Analysis and Discriminative Modeling of Filled Pauses for Spontaneous Speech Recognition". *Journal of VLSI Signal Process. Syst.* 36, 2-3 (Feb. 2004), 91-104.

Žgank, A., Kačič, Z., Horvat, B, (2001). "Large vocabulary continuous speech recognizer for Slovenian language". *Lecture notes computer science, 2001*, pp. 242-248, Springer Verlag.

Žgank, A., M. Rojc, B. Kotnik, D. Vlaj, M. Sepesy Maučec, T. Rotovnik, Z. Kačič, A. Zögling Markuš, B. Horvat, (2002). Govorno voden informacijski portal LentInfo – predhodna analiza rezultatov. *Jezikovne tehnologije 2002*, Inštitut Jožef Stefan, Ljubljana.

Žgank, A., Kačič Z., Horvat, B., (2003). "Data driven generation of broad classes for decision tree construction in acoustic modeling", *In: EUROSPEECH 2003*, Geneva, Switzerland, 2505-2508.

Žgank, A., Horvat, B., Kačič Z., (2005). "Data-driven generation of phonetic broad classes, based on phoneme confusion matrix similarity". *Speech Communication* 47(3): 379-393.

Žgank, Andrej, Verdonik, Darinka, Zögling Markuš, Aleksandra, Kačič, Zdravko, (2005). "BNSI Slovenian broadcast news database - speech and text corpus", *9th European conference on speech communication and technology*, September, 4-8, Lisbon, Portugal. Interspeech Lisboa 2005, Portugal.

Žgank, A., Rotovnik, T., Maučec Sepesy, M., Kačič Z., (2006). "Basic Structure of the UMB Slovenian Broadcast News Transcription System", *Proc. IS-LTC Conference*, Ljubljana, Slovenia.

Žgank, A., Verdonik, D., Zögling Markuš, A., Kačič Z., (2006). SINOD - Slovenian non-native speech database. *Proc. LREC 2006*, Genova, Italy.

Žgank, A., Rotovnik, T., Sepesy Maučec, M., (2008) Slovenian Spontaneous Speech Recognition and Acoustic Modeling of Filled Pauses and Onomatopoeas, WSEAS TRANSACTIONS on SIGNAL PROCESSING, Issue 7, Volume 4, July 2008.

6

Non-native Pronunciation Variation Modeling for Automatic Speech Recognition

Mina Kim[1], Yoo Rhee Oh[2] and Hong Kook Kim[2]
[1]*Mobile Communication Department, LG Electronics*
[2]*School of Information and Communications*
Gwangju Institute of Science and Technology
Korea

1. Introduction

Communication using speech is inherently natural, with this ability of communication unconsciously acquired in a step-by-step manner throughout life. In order to explore the benefits of speech communication in devices, there have been many research works performed over the past several decades. As a result, automatic speech recognition (ASR) systems have been deployed in a range of applications, including automatic reservation systems, dictation systems, navigation systems, etc.

Due to increasing globalization, the need for effective interlingual communication has also been growing. However, because of the fact that most people tend to speak foreign languages with variant or influent pronunciations, this has led to an increasing demand for the development of non-native ASR systems (Goronzy et al., 2001). In other words, a conventional ASR system is optimized with native speech; however, non-native speech has different characteristics from native speech. That is, non-native speech tends to reflect the pronunciations or syntactic characteristics of the mother tongue of the non-native speakers, as well as the wide range of fluencies among non-native speakers. Therefore, the performance of an ASR system evaluated using non-native speech tends to severely degrade when compared to that of native speech due to the mismatch between the native training data and the non-native test data (Compernolle, 2001). A simple way to improve the performance of an ASR system for non-native speech would be to train the ASR system using a non-native speech database, though in reality the number of non-native speech samples available for this task is not currently sufficient to train an ASR system. Thus, techniques for improving non-native ASR performance using only small amount of non-native speech are required.

There have been three major approaches for handling non-native speech for ASR: acoustic modeling, language modeling, and pronunciation modeling approaches. First, acoustic modeling approaches find pronunciation differences and transform and/or adapt acoustic models to include the effects of non-native speech (Gruhn et al., 2004; Morgan, 2004; Steidl et al., 2004). Second, language modeling approaches deal with the grammatical effects or speaking style of non-native speech (Bellegarda, 2001). Third, pronunciation modeling approaches derive pronunciation variant rules from non-native speech and apply the derived rules to pronunciation models for non-native speech (Amdal et al., 2000; Fosler-Lussier, 1999; Goronzy et al., 2004; Gruhn et al., 2004; Raux, 2004; Strik et al., 1999).

The remainder of this chapter is organized as follows. In Section 2, an overview of non-native speech recognition is investigated. After that, acoustic modeling, language modeling, and pronunciation modeling approaches are explained in Sections 3, 4, and 5, respectively. Then, a new pronunciation modeling method is proposed in Section 6 as a means of improving the performance of non-native speech recognition. In addition, the performance of a non-native ASR system adopting the proposed method is evaluated and compared to that employing conventional pronunciation model adaptation methods. Finally, we conclude our findings in Section 7.

2. Overview of non-native speech recognition

Recently, speech recognition technology has become more familiar in our lives (Goronzy et al., 2001), as numerous applications are increasingly adopting speech recognition systems. For example, voice dialing is possible based on either a user stating a name or a number, dictation systems are relatively common, and there are a number of voice-enabled automatic response systems available. However, when these ASR systems are used by non-native speakers, the performance of the system can rapidly degrade because of the mismatches between the native training data and the non-native test data (Compernolle, 2001).

Previously, several works have investigated the characteristics of non-native speech and the effect of non-native speech on ASR performance, some of which tried to explore the differences in characteristics between native and non-native speakers. For examples, the authors of (Sidaras et al., 2009) demonstrated that the duration and the first and second formant frequencies of English vowels spoken by Spanish speakers had different characteristics from those of native English speakers. Moreover, it was found that Spanish-accented English was perceived better when the listeners were trained with this form of English. Similarly, it was noticed that the tongue location of the English vowels by non-native speakers had different characteristics from that of native speakers (Wade et al., 2007). In addition, according to the work in (Alotaibi et al., 2010), unique consonants existed in some languages, such as four emphatic consonants of Arabic, and these unfamiliar consonants were found to be hard to perceive by non-native speakers. It was then found that when non-native speakers pronounced words containing these unfamiliar consonants, degradation of ASR performance could occur.

Other researchers have attempted to compare the ASR performance of both native and non-native speech. In (Wang et al., 2003), it was shown that the word error rate (WER) of an English ASR system by German speakers was 49.3% whereas that of native English speakers was 16.2%. Moreover, in (Steidl et al., 2004), an ASR system trained by German speakers provided WERs of 18.5% and 34.0% when tested by native German speakers and English speakers, respectively. However, when the same ASR system was trained by English speakers but tested by German speakers, the WER increased from 35.0% to 65.6%. Based on these previous works, it is evident that adjusting for different pronunciation characteristics between native and non-native speakers is crucial for improving the ASR performance of non-native speech.

In order to improve the ASR performance for non-native speech, we first need to prepare a non-native speech database to train the ASR system or adjust the system for non-native speech; then, each component of the ASR system can be adjusted for non-native speech. Depending on which ASR component is adapted or modified for non-native speech, we can classify the techniques developed for non-native speech as shown in Fig. 1. In brief, a typical

ASR system is composed of a front-end for extracting acoustic feature, acoustic models for representing recognition units with the acoustic features, a language model for covering language-specific grammar or syntax, and a pronunciation model for handling the phonology, phonotactics, or phonetics of the target language. Therefore, different techniques can deal with non-native ASR issues from acoustic modeling, language modeling, or pronunciation modeling points of view. In addition, it is also important to consider how to transform or compensate for acoustic features extracted from non-native speech into native speech. It is suggested here that to further improve ASR performance, a hybrid modeling approach can be used, one that combines some or all of the approaches mentioned above.

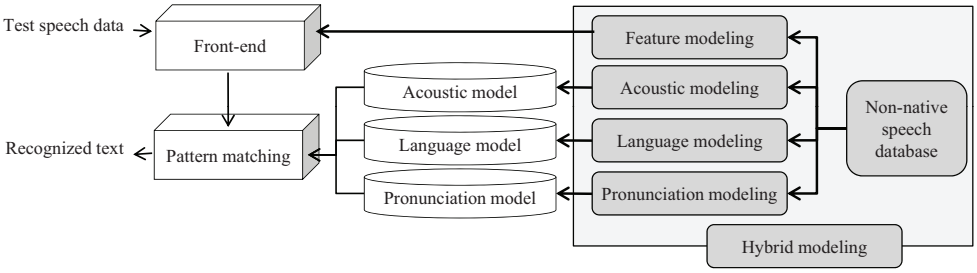


Fig. 1. Classification of techniques applied to non-native ASR.

1. Non-native speech database design
   In order to develop a non-native ASR system and investigate the characteristics of non-native speech, we first require non-native speech databases; Raab et al. (Raab et al., 2007) have previously reviewed such non-native speech databases.
2. Acoustic modeling approach
   Acoustic modeling approaches are used to adjust acoustic models and thereby improve the recognition performance of non-native speech (Gruhn et al., 2004; Morgan, 2004; Steidl et al., 2004). A simple way of adjusting acoustic models is to train them using a large amount of non-native speech. However, in practice it is rather difficult to collect a sufficient amount of non-native speech; therefore, acoustic models are usually adapted via a conventional acoustic model adaptation method, such as maximum likelihood linear regression (MLLR) and/or maximum a posteriori (MAP) methods (Yang et al., 2004). As an alternative, the acoustic models adjusted for non-native speech can also be obtained by interpolating the acoustic models for native speech and the acoustic models for the mother tongue (Steidl et al., 2004; Tan et al., 2007). In other words, the acoustic models trained with two different languages are combined to obtain the acoustic models for non-native speech. However, the most popular way of obtaining the adjusted acoustic models is to apply an adaptation technique with only small amount of adaptation data for non-native speech (Liu et al., 2008; Oh et al., 2007; 2009).
3. Language modeling approach
   Language modeling approaches deal with the grammatical effects or speaking styles of non-native speech, since non-native speakers tend to make a different sentence structure from native speakers (Bellegarda, 2001). However, there are relatively few research works in this area, compared to either the acoustic modeling approaches or the pronunciation modeling approaches (Huang et al., 2008; Raux et al., 2004; Steidl et al., 2004).

4. Pronunciation modeling approach
   Pronunciation modeling approaches first derive pronunciation variants from non-native speakers and then apply them to the pronunciation models for non-native speech. Usually, the variant pronunciations for each word are added to the pronunciation models, which is similar to a multiple pronunciation dictionary approach (Amdal et al., 2000; Fosler-Lussier, 1999; Goronzy et al., 2004; Gruhn et al., 2004; Raux, 2004; Strik et al., 1999).
   The pronunciation variants from non-native speakers can be derived by either knowledge-based or data-driven approaches (Strik et al., 1999). Note that knowledge-based approaches are based on linguistics or phonetic knowledge (Schaden, 2003; Tajchman et al., 1995; Wiseman et al., 1998), whereas data-driven approaches automatically derive pronunciation variants from non-native speech data and can be further classified into either a direct method (Amdal et al., 2000; Fosler-Lussier, 1999; Strik et al., 1999) or an indirect method (Amdal et al., 2000; Fosler-Lussier, 1999; Goronzy et al., 2004; Svendsen, 2004; Wolff et al., 2001).
   If many pronunciation variants are derived, the adapted pronunciation model becomes enlarged, resulting in performance degradation of the ASR system due to the fact that confusability in the pronunciation model is increased. Thus, several confusability reduction methods have also been proposed (Amdal et al., 2000; Hernandez-Abrego et al., 2004; Tsai et al., 2002).
5. Hybrid modeling approach
   Hybrid modeling approaches combine several modeling approaches, as described above, to further improve the performance of non-native ASR. In other words, acoustic or pronunciation modeling approaches can be combined in an MLLR and/or MAP adaptation framework (Goronzy et al., 2004; He et al., 2003; Liu et al., 2008; Oh et al., 2007; 2010; Tan et al., 2007). In particular, Bouselmi et al. (Bouselmi et al., 2007) proposed several combination schemes for pronunciation and MLLR/MAP acoustic model adaptations. On the other hand, pronunciation variant rules were decomposed into either pronunciation or acoustic variants (Oh et al., 2008). After that, pronunciation and acoustic model adaptations were applied to pronunciation and acoustic variants, respectively.
6. Feature-domain approach
   The feature-domain approach applies a feature adaptation method to compensate for mismatches between training and test conditions; the acoustic models are trained using native speech, but are tested using non-native speech. For example, Oh and Kim (Oh et al., 2010) applied a feature-space MLLR (fMLLR) adaptation with smoothing techniques to non-native ASR.

The next three sections will provide more detailed descriptions of the acoustic, language, and pronunciation modeling approaches.

3. Acoustic modeling approach

Because of the limited non-native speech database mentioned in Section 2, interpolating or adapting existing acoustic models using a small amount of non-native speech data is preferred, rather than attempting to train new acoustic models using large amounts of non-native speech data. Thus, in this section we introduce a number of acoustic modeling approaches in attempts to improve the performance of non-native ASR by using only a limited amount of non-native speech data.

As an effort to adapt acoustic models, several interpolation methods have been proposed, where two sets of acoustic models, the acoustic models trained with the target language and the acoustic models trained with the mother tongue of native speakers, are combined (Matsunaga et al., 2003; Steidl et al., 2004; Tan et al., 2007). Contrary to interpolating acoustic models, phone acoustic models of the target language were modified by adding an alternative path to the corresponding mother tongue phone acoustic models of non-native speakers (Bartkova et al., 2006; Bouselmi et al., 2006). Finally, the acoustic models were adapted by using non-native adaptation data via either an algorithm dedicated to non-native ASR or a conventional speaker adaptation method (Liu et al., 2008; Oh et al., 2007; 2009).

3.1 Retraining method

A retraining method generates non-native acoustic models by using a large amount of non-native speech data or retrains native acoustic models by using a moderately large sample of non-native speech data. These types of retraining methods are very simple but have several drawbacks, such as the following.

First, retraining methods require a large amount of non-native speech and their corresponding transcription data; however, these data are usually limited in quantity. Second, the transcriptions of a non-native speech database cannot be automatically generated since some non-native speech data contain various unpredictable pronunciations and structural errors. Third, the performance of ASR systems employing the retrained non-native acoustic models tends to drastically degrade for native speech (Oh et al., 2007).

For these reasons, several alternative methods have been proposed, which either interpolate the native and non-native acoustic models or adapt the native acoustic models based on a relatively small non-native database.

3.2 Interpolation method

In this subsection, we explain several interpolation methods, classified as either: 1) interpolation of native acoustic models of target language using non-native speech data (Steidl et al., 2004), and 2) interpolation of native acoustic models of target language based on native acoustic models of the mother tongue of non-native speakers (Tan et al., 2007).

3.2.1 Use of target language acoustic models

In this category, the acoustic model interpolation method is based on two assumptions. First, each non-native pronunciation has at least one similar native pronunciation in the target language, stemming from the fact that most languages have very similar phone inventories. Second, the native acoustic models of the target language are sufficient for adapting acoustic models for non-native speech.

The procedure of the acoustic model interpolation method is as follows:

Step 1. Generation of transcriptions based on native acoustic models

Each non-native utterance in a development set is recognized by the native acoustic models of the target language, which then automatically generates the transcriptions. According to the recognition results, each pronunciation in the lexicon is replaced by the recognized monophone such that highly specialized pronunciations in the lexicon are adapted.

Step 2. Selection of optimal interpolation partners

To select the optimal *K*-1 partners for acoustic model interpolation, each candidate partner is first interpolated based on the state of a hidden Markov model (HMM) of the target language, as shown in Eq. (1). Next, an *N*-best list of candidate partners is evaluated, and the first *K*-1 candidate partners are then selected from the *N*-best list.

Step 3. Interpolation of selected acoustic models

Since semi-continuous HMMs share the same set of output density probabilities, only the interpolation weights and the corresponding transition probabilities need to be adjusted in order to interpolate native acoustic models of the target language for non-native acoustic models. When there are *K*-1 interpolation partners for the state s_i of an HMM, the mixture weight $c_{i,m}$ of a state s_i of the HMM is adjusted as $\hat{c}_{i,m}$, based on the following equation:

$$\forall m \quad \hat{c}_{i,m} = \rho_1 \cdot c_{i_1,m} + \ldots + \rho_K \cdot c_{i_K,m} \tag{1}$$

where s_{i_1} represents s_i, and $s_{i_1},\cdots,s_{i_K}$ indicate the states of the corresponding interpolation partners of the state s_i. $\rightarrow c_{i_1}$ represents the mixture weight of s_i, and $c_{i_1},\cdots,c_{i_K}$ indicate the mixture weights of the states of the corresponding interpolation partners of the state s_i. In addition, ρ_1, …, ρ_K are the interpolation weights.

The interpolation weights indicate the probability from the original state s_i to the states of the corresponding interpolation partner and can be estimated using an expectation-maximization (EM) algorithm. After the interpolation weights are estimated, the corresponding transition probabilities can be determined in a similar manner.

3.2.2 Combined use of target language and mother tongue acoustic models

Tan and Besacier (Tan et al., 2007) proposed three interpolation methods based on the use of both the target language acoustic models and the mother tongue acoustic models of non-native speakers, which include 1) manual interpolation, 2) weighted least square based interpolation, and 3) eigenvoice based interpolation. The three acoustic model interpolation methods consist of two identical steps for preprocessing and one different step for the acoustic model interpolation.

Step 1. Investigation of phoneme mapping information

The mapping information on the phoneme substitutions for non-native speech is investigated using both the knowledge-based and the data-driven approaches.

- Knowledge-based approach
  Phoneme substitutions from the mother tongue of non-native speakers to the target language are first examined based on the international phonetic alphabet (IPA) tables (International Phonetic Association, 1999).
- Data-driven approach
  For a phoneme whose substitution information is not known from the IPA tables, a data-driven approach is applied using a phoneme confusion matrix. In other words, a forced alignment is first performed based on the target language acoustic models for each non-native utterance in a development set. Then, phoneme recognition is also performed using the mother tongue acoustic models for each non-native utterance. Next, the two phoneme sequences are aligned using time information in order to generate the phoneme confusion matrix. From the generated confusion matrix, the mapped phoneme having the highest probability is selected as the phoneme substitution for each phoneme.

Step 2. Regeneration of mother tongue acoustic models of non-native speakers

Before interpolating acoustic models, the mother tongue acoustic models of non-native speakers are reconstructed from the target language acoustic models in order to match the configuration of the target language acoustic models. For this task, the pronunciation dictionary of the mother tongue of non-native speakers is first modified using the investigated mapping information. The mother tongue acoustic models of non-native speakers are then reconstructed from the target language acoustic models by performing MLLR and MAP adaptations based on the speech corpus of the mother tongue of non-native speakers and the modified pronunciation dictionary.

- In the cases of manual and weighted least square based interpolations
  The mother tongue acoustic models of non-native speakers are reconstructed from the target language acoustic models by performing MLLR and MAP adaptations based on all the speech data of the mother tongue of non-native speakers and the modified pronunciation dictionary. In other words, speaker-independent acoustic models of the mother tongue are obtained as the mother tongue acoustic models.
- In the case of eigenvoice based interpolation
  For each native speaker of a speech training corpus, the target language acoustic models are reconstructed by performing MLLR and MAP adaptations using a subset of the speech corpus of the target language for the corresponding speaker and the original pronunciation dictionary. In other words, several sets of speaker-dependent acoustic models of the target language are obtained.
  Next, for each non-native speaker in a development speech corpus, the mother tongue acoustic models of non-native speakers are reconstructed from the target language acoustic models by performing MLLR and MAP adaptations using a subset of the speech corpus of the mother tongue for the corresponding speaker and the modified pronunciation dictionary. As a result, several sets of speaker-dependent acoustic models for the mother tongue are obtained.

Step 3.a. Manual interpolation of acoustic models

For the non-native acoustic models ($p_{interpolated}$) of a phoneme, the target language acoustic models ($p_{target\_language}$) for the phoneme are then interpolated based on the mother tongue acoustic models (p_{mother_tongue}) of the corresponding mapping phoneme, using the equation of

$$p_{interpolated} = w \cdot p_{target\_language} + (1 - w) \cdot p_{mother\_tongue} \tag{2}$$

where w ($0 \le w \le 1$) indicates an interpolation weight. In this method, the interpolation weight (w) is manually determined by experiments; this method is appropriate in the case that no non-native speech is available.

Step 3.b. Weighted least square based interpolation of acoustic models

If the non-native adaptation data are available, the interpolation weight can be predicted using the weighted least square. In other words, Eq. (2) can be rewritten as

$$A \cdot x = (p_{target\_language} \;\; p_{mother\_tongue}) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = (p_{interpolated}) = b \tag{3}$$

where b is calculated as the speaker means obtained by a forced-alignment with the non-native adaptation data on the target language acoustic models.

Given A and b as in Eq. (3), the interpolation weight vector x can then be solved by using the weighted least square as

$$A^T \cdot \Sigma^{-1} \cdot A \cdot x = A^T \cdot \Sigma^{-1} \cdot b \tag{4}$$

where the speaker variance Σ is a weight since each mean does not have the same weight.

Step 3.c. Eigenvoice based interpolation of acoustic models

From all the generated sets of speaker-dependent acoustic models for the target language and the mother tongue, the means act as supervectors for creating a non-native space for eigenvoice based interpolation. Thus, a subset of these eigenvectors is selected for the interpolation.

3.3 Adaptation method

In order to compensate for mismatches between the native training data and the non-native test data of the target language, the native acoustic models of the target language are adapted using non-native speech such that the ASR performance for non-native speech can be improved. As a simple adaptation, traditional acoustic model adaptation methods, which are widely used for speaker adaptations or noise-robust ASR, can be applied. However, traditional MLLR and/or MAP adaptation methods adapt only speaker or environmental variability, not pronunciation variability from non-native speakers. Hence, this subsection focuses on acoustic model adaptation methods for handling pronunciation variability from non-native speakers (Oh et al., 2007; 2009).

3.3.1 Modified decision-tree based state-clustering method

The modified decision-tree based state-clustering method is performed in a decision-tree based state-tying step during construction of the acoustic models. The main procedure of the modified decision-tree based state-clustering method is as follows:

Step 1. Analysis of pronunciation variability of non-native speakers

Since the modified decision-tree based state-clustering method is based on the pronunciation variability of non-native speech, this pronunciation variability is first investigated in an indirect data-driven method that will be further explained in Section 6. In brief, for each utterance in a non-native development set, phoneme recognition is performed and then an N-best list of phoneme sequences is obtained. Next, the phoneme rule patterns that are derived from the recognized N-best lists are applied to a decision tree, C4.5 (Quinlan, 1993). As a result, the pronunciation variant rules are generated.

Step 2. Decomposition of pronunciation variability of non-native speakers

Among the derived pronunciation variant rules, *acoustic variants* are selected in the case that the default class ($phoneme_{default}$) of the pronunciation variant rule has a different phoneme from a target phoneme ($phoneme_{target}$). Other pronunciation variant rules are then determined as *pronunciation variants*. Note that only the acoustic variants are applied to the modified decision-tree based state-clustering method.

The acoustic and pronunciation variants can be briefly explained as follows:

- Acoustic variants, $phoneme_{variant_{acoustic}}$

  Acoustic variants are named since the pronunciation variant rules are applied in the acoustic modeling. In addition, it is assumed that the variants occurred due to

the different pronunciation characteristics between the target language and the non-native speaker's mother tongue. These acoustic variants can be placed in any context and thus they are also referred to as *context-independent variants*. For this reason, acoustic modeling is more appropriate than pronunciation modeling since pronunciation modeling adds variant pronunciations for each corresponding context and thereby increases the confusability.

- Pronunciation variants, $phoneme_{variant_{pronunciation}}$
  Pronunciation variants are named since the pronunciation variant rules are applied in the pronunciation modeling. In addition, it is assumed that the variants are due to the co-articulation effect. In the model, these pronunciation variants would be placed in a specific context with the left two phonemes and the right two phonemes, and thus they are also referred to as *context-dependent variants*. Consequently, pronunciation modeling can more properly handle the pronunciation variants by adding the corresponding variant pronunciations of each word.

Step 3. Adaptation in the state-tying step of acoustic model construction

The acoustic model adaptation is performed in the decision-tree based state-tying step of acoustic model construction using the acoustic variants. For a phoneme having no acoustic variants, a traditional state-tying step is applied, in which a decision tree for each target phoneme ($phoneme_{target}$) is utilized based on the states of the triphone acoustic models where the central phone of the triphone has the $phoneme_{target}$. However, for a phoneme having acoustic variants, a decision tree for each $phoneme_{target}$ is utilized by using the states of the triphone acoustic models in which the central phone of the triphone has either $phoneme_{target}$ or $phoneme_{variant_{acoustic}}$.

3.3.2 Modified MLLR adaptation method

A traditional MLLR adaptation method is commonly used for speaker or environment variants; however, the MLLR adaptation should be modified for non-native ASR (Oh et al., 2009). In other words, an MLLR/MAP adaptation for triphone models having pronunciation variations is performed to handle the pronunciation variability of non-native speakers. The main procedure of the modified MLLR adaptation method is as follows:

Step 1. Acquisition of pronunciation variations of non-native speech

The pronunciation variations of non-native speech are generated in an indirect data-driven approach, as will be explained in Section 6. Then, the only acoustic variants are selected by investigating the pronunciation variant rules in which the default class has a different phoneme as the target phoneme, as described in Section 3.3.1.

Step 2. Generation of regression classes

In this step, two separate sets of regression classes are generated; *overall regression classes* for the characteristics of non-native speakers or environments, and *pronunciation variation regression classes* for the pronunciation variations of non-native speech.

- For the overall regression classes
  All the acoustic models of the target language are pooled on the root node of a regression class tree and the overall regression classes are then generated by splitting the regression class tree to adapt the acoustic models of the target language for the characteristics of non-native speakers or environments.
- For the pronunciation variation regression classes
  Pronunciation variation regression classes are generated for each pronunciation having acoustic variants. That is, the acoustic models for both the target

pronunciation and the corresponding variant pronunciations are pooled on the root node of a regression class tree, and the pronunciation variation regression classes for the target pronunciation are then generated by splitting the regression class tree such that the acoustic models of the target language are adapted for the pronunciation variations of non-native speech.

In order to generate a regression class, the acoustic models pooled on the root node of a regression class tree are first split based on the criterion of the centroid splitting algorithm, using the Euclidean distance measure (Young et al., 2002). Then, each regression class is identified by using the acoustic models clustered on the leaf node of the regression class tree.

Step 3. Adaptation of acoustic models using MLLR and MAP adaptation methods

It is known that the combination of MLLR and MAP adaptations can further improve the ASR performance of non-native speech, as opposed to using either only the MLLR or MAP adaptations (Goronzy et al., 2004; He et al., 2003; Tan et al., 2007). Therefore, a second-pass adaptation method using both the MLLR and MAP adaptations is performed in order to adapt the acoustic models of the target language (Oh et al., 2009). In other words, for each regression class, the corresponding MLLR transformation matrix is first estimated via an EM algorithm based on the non-native adaptation data. Then, the adapted acoustic models are generated by applying a MAP adaptation with the non-native adaptation data and the estimated MLLR transformation matrix.

Step 4. Reconfiguration of the adapted acoustic models

Since one set of adapted acoustic models from the overall regression classes and several different sets of adapted acoustic models from the pronunciation variation regression classes are generated in Step 3) of this subsection, a single set of adapted acoustic models should be selected. To this end, for each pronunciation variation, the corresponding models in the adapted acoustic models from the overall regression classes are replaced by the acoustic models adapted by the corresponding pronunciation variation regression class. Accordingly, the reconfigured acoustic models can cover the characteristics of non-native speakers or environments as well as the pronunciation variations of non-native speech.

4. Language modeling approach

Language modeling approaches are associated with the different speaking styles or the grammatical effects of non-native speech. When compared to either the acoustic or pronunciation modeling approaches, there have been few research works reported on language modeling. Nevertheless, in this section, we explain the language modeling method for continuous word speech recognition and for pronunciation grammar (Huang et al., 2008; Raux et al., 2004; Steidl et al., 2004).

4.1 Interpolation with non-native language model

Non-native speakers tend to make different sentence structures from native speakers due to the syntactic characteristics of the mother tongue of non-native speakers. For handling such syntactic differences of non-native speech, Steidl et al. (Steidl et al., 2004) employed an adapted language model by combining the original native language model and the non-native language model. The non-native language model was generated by using the transliteration of a non-native speech database. In addition, Raux and Eskenazi (Raux et al.,

2004) generated a non-native language model for a language learning system having both native and non-native speech data. It was shown from subsequent experiments that both methods improved the recognition performance when compared to the native language model.

4.2 Unsupervised pronunciation grammar growing

Huang et al. (Huang et al., 2008) proposed an unsupervised pronunciation grammar growing method in order to obtain the grammar of the pronunciation variations of non-native speakers and to generate the pronunciation models for non-native speech. The method consisted of two steps: the construction of a pronunciation variation graph and the generation of the non-native grammar from the pronunciation variation graph.

The main procedure of the unsupervised pronunciation grammar growing method is as follows:

Step 1. Construction of a pronunciation variation graph

A pronunciation variation graph for a word starts with all the possible pronunciation variations including insertions, deletions, and substitutions. Thus, a huge search space is required for the pronunciation variation graph of a word. In the graph, a node indicates the possible pronunciation and an edge represents the possible transition between pronunciations. In order to reduce the search space of the graph, the possible pronunciations and transitions for a substitution are first constrained within the broad class information defined by linguistic experts. Next, the possible paths remaining for the pronunciation variations are evaluated by calculating the posterior probabilities of each phone pair (ph_{start}, ph_{end}) using the equation,

$$\frac{1}{N}\sum_{i}^{N} p(x_i \left| \lambda_{ph_{end}} \right.) = \frac{1}{N}\sum_{i}^{N} \frac{1}{\sqrt{(2\pi)^d \left| \Sigma_{ph_{end}} \right|}} \exp[-\frac{1}{2}(x_i - \mu_{ph_{end}})^T \Sigma_{ph_{end}}^{-1} (x_i - \mu_{ph_{end}})] \quad (5)$$

where x_i, N, and d indicate the i-th observation feature vector corresponding to ph_{start} in a training speech corpus, the number of observation feature vectors corresponding to ph_{start} in the training speech corpus, and the dimension of the observation feature vector, respectively. In addition, $\lambda_{ph_{end}}$, $\mu_{ph_{end}}$, and $\Sigma_{ph_{end}}$ represent the acoustic model, the mean vector, and the covariance matrix for the phone ph_{end}. In the experiment, paths that are greater than a predefined threshold remain in the pronunciation variation graph.

Next, the possible left-context and right-context dependent pronunciations are generated using both a target language pronunciation dictionary and a mother tongue pronunciation dictionary. Then, only the possible paths having context dependent pronunciations are extracted.

Step 2. Generation of non-native grammar

By using the constructed pronunciation variation graph, speech recognition is first performed and the pronunciation variation grammar is then optimized by removing the pronunciations that are incorrectly recognized or have unusual variants based on the recognition confidence and support score. Here, the word-level generalized posterior probability and the occurrence frequency of the pronunciation variation are used as the recognition confidence and the support score, respectively. The finally optimized pronunciation variation grammar is subsequently used to generate the multiple pronunciation dictionary for non-native speakers.

5. Pronunciation modeling approach

There are two approaches pertaining to pronunciation model adaptations for non-native speech: a knowledge-based approach and a data-driven approach (Strik et al., 1999). A knowledge-based approach uses pronunciation rules from phonological knowledge and develops a pronunciation dictionary based on the pronunciation rules. In the case of a data-driven approach, phonological rules for pronunciation adaptation are automatically generated from non-native speech and transcription data; as such, a subdivision into direct and indirect data-driven methods can be applied.

5.1 Knowledge-based method

In a knowledge-based method, phonologically obtained pronunciation rules are used to transform a baseform into a pronunciation variant. For example, the phonological rule

$$vowel + /b/ + /d/ \rightarrow vowel + /b/ + /D/ \tag{6}$$

is used to transform a consonant $/d/$ followed by a consonant $/b/$ into a fortis consonant $/D/$ in Korean. The phonological rules are derived based on linguistical and phonological knowledge according to known pronunciation variations of speech. Then, the phonological rules are applied to baseforms in a pronunciation dictionary.

As representatives of knowledge-based approaches, pronunciation rules from phonological knowledge were previously generated to develop a pronunciation dictionary based on pronunciation rules (Tajchman et al., 1995; Wiseman et al., 1998). Also, Schaden (Schaden, 2003) transformed canonical phonetic dictionaries of the target language into adapted dictionaries in order to model prototypical foreign-accented pronunciation variants.

5.2 Data-driven method

The primary advantage of the knowledge-based approach is that it can be applied to all corpora and especially to new words that are not introduced in the ASR system. However, a notable drawback of the approach is in that the rules are often very general, resulting in too many variants in the pronunciation dictionary, thereby increasing the confusability of pronunciation variations. Moreover, it should be noted that even if this approach is applied to an ASR system, it is unlikely that all aspects of non-native speech could be covered.

In order to compensate for such drawbacks of the knowledge-based approach, pronunciation variations are derived from speech signals in data-driven methods. Data-driven methods can be further classified into direct data-driven or indirect data-driven approaches. The direct data-driven approach derives pronunciation variants depending on pronunciation training databases, as proposed in (Amdal et al., 2000; Fosler-Lussier, 1999; Strik et al., 1999). When an ASR system employs the adapted pronunciation dictionary using a direct data-driven approach, some unseen words might appear during ASR testing. Thus, such a mismatch condition in the pronunciation model between ASR training and testing could degrade the performance of an ASR system.

On the other hand, an indirect data-driven method investigates pronunciation variability from the speech training data, derives the variant rules, and applies the variant rules in the ASR pronunciation dictionary to compensate for the variability (Amdal et al., 2000; Fosler-Lussier, 1999; Goronzy et al., 2004; Svendsen, 2004; Wolff et al., 2001). For example, pronunciation rules were derived using the speech training data, which in turn could be applied to generate one or

more baseforms of any vocabulary word in the pronunciation dictionary (Svendsen, 2004). In addition, variants were derived using a phoneme recognizer such that pronunciation rules could be constructed using a decision tree (Fosler-Lussier, 1999). Confidence measures were then used to select only the most reliable variants from among all the recognized variants; a similar approach was applied in the Verbmobil project reported in (Wolff et al., 2001). As another example, non-native speech was first examined using a phoneme recognizer to determine variants, and then variants caused by recognition errors were removed based on the statistics pertaining to the co-occurrences of phonemes (Amdal et al., 2000). In this way, Goronzy et al. (Goronzy et al., 2004) used an English phoneme recognizer to generate English pronunciations for German words and used decision trees that were able to predict English-accented variants from German canonical transcriptions.

5.3 Confusability reduction of pronunciation dictionary

As described above, data-driven methods adapt a pronunciation dictionary after building variant rules from the derived pronunciation variants, whereas a knowledge-based method derives variant rules based on phonological and phonetic knowledge, and then adds alternatives of pronunciation variants into the pronunciation dictionary. However, the adapted pronunciation dictionary can have more than one element corresponding to a word in the pronunciation dictionary. Therefore, the system memory size must be increased in order to store the pronunciation dictionary, which also increases the computational complexity and results in a longer decoding time for ASR. It was also observed that adding pronunciation variants to the pronunciation dictionary increases the confusability, and that a large increase in confusability is probably one reason for only small improvements or even deteriorations of ASR performance (Tsai et al., 2002). By appropriately selecting the pronunciation variations, the confusability would be reduced. In order to mitigate this problem, several approaches have been previously reported, which will be discussed in Section 6.2.

6. Pronunciation model adaptation based on multiple pronunciation dictionary

In this section, we describe a new pronunciation model adaptation method and an optimization method of the adapted pronunciation models proposed in (Kim et al., 2008). In particular, Section 6.1 describes the proposed pronunciation adaptation method based on an indirect data-driven approach that adapts a pronunciation dictionary after building the variant rules from the derived pronunciation variants, resulting in a *multiple pronunciation dictionary*. This dictionary can have more than one element corresponding to a word in the pronunciation dictionary. Thus, a size optimization method of the multiple pronunciation dictionary is also proposed in Section 6.2, in which some confusable pronunciation variants in the pronunciation dictionary are removed. Finally, in Section 6.3, the performance of a non-native ASR system employing the proposed method is evaluated and compared with that using a conventional pronunciation model adaptation method.

6.1 Multiple pronunciation dictionary

Fig. 2 shows the main procedure of the proposed pronunciation variation modeling method based on an indirect data-driven approach that is applied to non-native speech. From the figure, the five steps of the procedure are as follows:

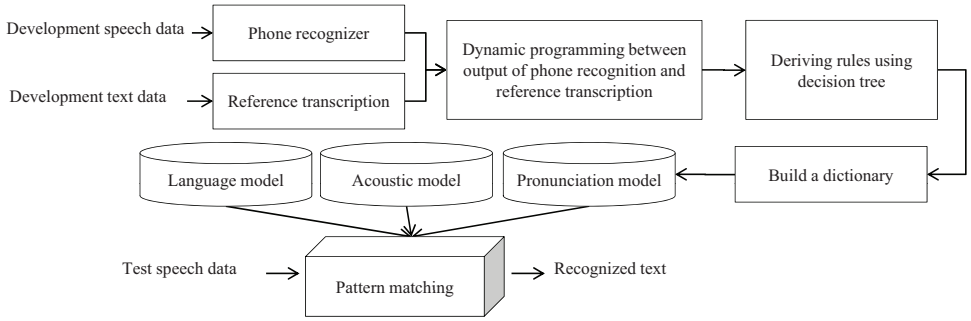


Fig. 2. Procedure of the proposed pronunciation variation modeling method based on an indirect data-driven approach applied to non-native ASR.

Step 1. Each utterance in a non-native development set is recognized using a phoneme recognizer.

Step 2. The recognized phoneme sequence is aligned using a dynamic programming algorithm based on the reference phoneme sequence transcribed by the native pronunciation dictionary, referred to as *reference transcription*.

Step 3. Using the alignment results of Step 2), variant phoneme patterns are obtained.

Step 4. Pronunciation variation rules are then derived from the variant phoneme patterns using a decision tree.

Step 5. Finally, pronunciation variations are generated from the pronunciation variation rules, allowing the pronunciation dictionary to be adapted for non-native ASR.

The details of each processing step are explained in further detail in the following subsections.

6.1.1 Phoneme recognition and aligned sequence

To derive the pronunciation rules, we first perform a phoneme recognition for each utterance in the non-native development set. As a result, we can obtain an N-best list of phoneme sequences for each utterance. However, there are no word boundaries in the list, which are required to differentiate inter-word pronunciation variations from cross-word pronunciation variations. To obtain these word boundaries, the recognized phoneme sequence is aligned on the basis of a dynamic programming algorithm and compared to the reference transcription with word boundaries.

From the alignment between the recognized phoneme sequence and the reference transcription, a rule pattern is obtained if the following condition is satisfied:

$$L_2 - L_1 - X + R_1 + R_2 \rightarrow Y \tag{7}$$

where X is a phoneme that is to be mapped into Y, and the left and right phonemes in the reference transcription are L_1 and L_2, and R_1 and R_2, respectively.

It is known from (Goronzy et al., 2004) that it is rather difficult to differentiate pronunciation variations from the substitution, deletion, and insertion errors incurred by phoneme recognition. Therefore, the recognition errors should be as small as possible; thus, three subsequent processes are applied to reduce these errors. First, we perform a Viterbi search based on the N-best lists. Second, we only extract a sentence or an isolated word included in

the development set if its phoneme recognition accuracy is over the predefined threshold. Third, if more than half of the neighboring phonemes of X in Eq. (7) are different from the neighboring phonemes of the target phoneme Y, this rule pattern is removed from the rule pattern set.

6.1.2 Decision-tree based rule derivation and pronunciation dictionary adaptation

Decision-tree based modeling is a popular method of deriving pronunciation variation rules (Fosler-Lussier, 1999; Wolff et al., 2001). Here, we use C4.5, a software extension of the basic ID3 algorithm designed by Quinlan (Quinlan, 1993). After the rule patterns are categorized by filtering errors, pronunciation variation rules are constructed by C4.5. Their attributes include the two left phonemes, L_1 and L_2, and the two right phonemes, R_1 and R_2, of the affected phoneme X. The output class is the target phoneme, where one decision tree is constructed for each phoneme. Next, each decision tree is converted into an equivalent set of the rules by tracing each path in the decision tree from the root node to each leaf node.
Next, a native pronunciation dictionary is adapted from these derived rules using C4.5, which results in a multiple pronunciation dictionary. For a more detailed description of adapting the pronunciation dictionary, refer to the work in (Kim et al., 2007).

6.2 Optimized multiple pronunciation dictionary

The size of the adapted multiple pronunciation dictionary could be much larger than that of the baseline pronunciation dictionary. As one solution to this problem, the confusability could be reduced by pruning the pronunciation variant rules based on either a rule probability, a rule probability using log likelihood, a decision tree, or another method. However, this approach does not take into account the interaction between words in a multiple pronunciation dictionary (Amdal et al., 2000). In other words, if a word is represented by several different phonetic sequences based on pronunciation variant rules and one of the sequences is similar to a phonetic sequence of another word, the confusability is further increased. Moreover, the confusability of words that have a smaller number of phonemes incurs errors in ASR systems (Hernandez-Abrego et al., 2004). Therefore, the number of phonemes in a word's sequence should be used as a measure of the confusability. In the following subsections, we propose a confusability measure and explain how the measure is applied to reduce the confusability in the multiple pronunciation dictionary of a non-native ASR system.

6.2.1 Confusability measure

Let M be a multiple pronunciation dictionary. It is assumed that the number of words in M is N_w and the i-th word, W_i, included in M, has $N_{p,i}$ pronunciation variants. Here, we denote $s_{i,j}$ as the j-th pronunciation variant belonging to the i-th word; i.e., $M=\{W_i \mid i = 1, \ldots, N_w\}$ and $W_i=\{s_{i,j} \mid j = 1, \ldots, N_{p,i}\}$. A confusability measure (CM) is then defined as

$$CM(s_{i,j}) = L(s_{i,j}) \cdot \min_{1\le k\le N_w, k\neq i, 1\le l\le N_{p,k}} \left\lfloor D(s_{i,j}, s_{k,l}) \cdot L(s_{k,l}) \right\rfloor \tag{8}$$

where $D(x,y)$ is the Levenshtein distance between x and y (Levenshtein, 1966). In addition, $L(x)$ is the number of phonemes of a pronunciation variant x, normalized by the maximal number of phonemes over all the pronunciations in M such that

$$l_{max} = max_{1 \le i \le N_w, 1 \le j \le N_{p,i}} \sharp(s_{i,j}) \quad (9)$$

where $\sharp(x)$ is defined as the number of phonemes in the pronunciation x. The goal of the proposed confusability measure defined in Eq. (8) is to detect pronunciation variants that are highly confusable so that ASR errors due to high similarities between the phonetic sequences of words in the multiple pronunciation dictionary can be reduced.
The normalized number of phonemes of a phonetic sequence x, is defined by

$$L(x) = l_x / l_{\max} \quad (10)$$

where $l_x = \sharp(x)$ and l_{max} is the maximum number of phonemes among all the sequences in M, as defined in Eq. (9). Eq. (10) contributes to the reduction of ASR errors because an ASR system tends to be more erroneous if the recognized word has a short phonetic sequence (Hernandez-Abrego et al., 2004). Therefore, the normalized number of a word's sequence can be used as a measure of the confusability.

6.2.2 Confusability reduction of multiple pronunciation dictionary

To reduce the confusability in the adapted multiple pronunciation dictionary, the confusability measure, defined in Eq. (8), for each pronunciation variant in the multiple pronunciation dictionary is first calculated. After that, all pronunciation variants except for the phonetic sequences obtained from the baseform are sorted according to their confusability measure scores. Finally, the pronunciation variants whose confusability measure scores are above a predefined threshold are used in constructing a pruned multiple pronunciation dictionary.

6.3 Speech recognition experiments

In order to evaluate the proposed pronunciation adaptation method, the baseline ASR system is first constructed. After that, we evaluate the performance of an ASR system using the pronunciation dictionary pruned by the proposed confusability reduction method, and compare it with that using the baseline pronunciation dictionary or the multiple pronunciation dictionary based on the indirect data-driven method.

6.3.1 Baseline ASR system

Especially, we want to develop a non-native ASR system that recognizes English spoken by Koreans. Thus, we need a training database spoken by native speakers to construct the baseline native ASR system. It is also required the native and non-native databases for developing and evaluating the non-native ASR system. In this subsection, we first describe the native and non-native databases. After that, we discuss how to construct each component of the baseline ASR system including ASR features, acoustic models, pronunciation and language models.

1. Training database
   As a training set for the baseline ASR system, we used a subset of the Wall Street Journal database (WSJ0) (Paul et al., 1992). The WSJ0 database was a 5000-word closed loop task for evaluating the performance of a large vocabulary continuous speech recognition (LVCSR) system. The training set consisted of 7,138 utterances recorded by a Sennheiser close-talking microphone and several far-field microphones, in which all utterances were sampled at a rate of 16 kHz.

2. Development and evaluation databases
   For developing and testing the proposed method, we used a subset of the Korean-Spoken English Corpus (K-SEC) (Rhee et al., 2004), comprised of English pronunciations spoken by both Korean and native English speakers. This database was divided into three parts: one was used for developing the pronunciation dictionary described in Section 6.1, and the others were evaluation subsets for both the baseline ASR system and an ASR system employing the proposed pronunciation modeling method. In other words, the two evaluation sets were comprised of utterances spoken by 49 Koreans and 7 native English speakers, respectively. The development set consisted of 11,125 isolated words spoken by 7 Koreans and 36 sentences by 98 Koreans, where each sentence had around 7 words. As a result, we had 7,299 isolated words and 3,123 continuous sentences for the development set. The two evaluation sets were made up of continuous sentences, in which each Korean or native speaker uttered 14 continuous sentences, resulting in a total of 146 words. In other words, we had 686 and 98 utterances for non-native speech and native speech, respectively.
3. Feature extraction
   For the baseline ASR system, we extracted 12 mel-frequency cepstral coefficients (MFCC) with logarithmic energy for every 10 ms analysis frame, and concatenated their first and second derivatives to obtain a 39-dimensional feature vector. During the training and testing, we applied a cepstral mean normalization to the feature vectors.
4. Acoustic models
   The acoustic models were based on 3-state left-to-right, context-dependent, 4-mixture, and cross-word triphone models, and they were trained using the HTK version 3.2 Toolkit (Young et al., 2002). All triphone models were expanded from 41 monophones, which included silence and pause models, and states of the triphone models were tied by employing a decision tree (Young et al., 1994). As a result, we had 9,655 physical triphones, 68,923 logical triphones, and 5,292 states, which was then referred to as the *baseline ASR system*.
5. Pronunciation and language models
   To develop a pronunciation dictionary, a back-off bigram language model was generated from a phoneme transcription of the training database, and the pronunciation dictionary was generated from a list of 41 phonemes with silence. In order to explore the behavior of pronunciation models based on the difference between the target language and the mother tongue, the pronunciation dictionary was only from the text of the test set. The pronunciation of each word was built from the CMU pronunciation dictionary (Weide, 1998) and any missing words from the CMU dictionary were transcribed manually. The pronunciation dictionary was comprised of 340 words, which was equal to the number of entries in the pronunciation dictionary. In addition, the relative ratio of the pronunciation dictionary size, defined as the average number of different pronunciations per word, was 1.

The performance of the baseline ASR system was tested using the two evaluation sets. Consequently, it was found that the average WERs of the baseline ASR system were 0.68% and 19.92% when the ASR system was tested by native speakers and by non-native speakers, respectively. This result confirmed the fact that performance of the ASR system tested by non-native speech could be exceedingly degraded.

Dictionary		WER (%)			Dictionary size (entry)	Relative ratio of dictionary	Real -time X
		Non-native	Native	Avg.			
a) Baseline		19.92	0.68	10.3	340	1	2.18
b) Multiple dictionary		21.96	0.68	11.32	512	1.51	3.5
c) Pruned multiple pronunciation dictionary							
Threshold	0	20.25	0.59	10.42	476	1.4	3.2
	0.1	**18.58**	**0.59**	**9.59**	**443**	**1.3**	**2.76**
	0.2	18.89	0.59	9.74	434	1.28	2.71
	0.3	18.93	0.59	9.76	428	1.26	2.67

Table 1. Performance comparison of an ASR system with a) the baseline pronunciation dictionary, b) a multiple pronunciation dictionary prior to reduction, and c) a pruned multiple pronunciation dictionary based on the proposed confusability reduction method. (Reprinted with permission from (Kim et al., 2008). Copyright IASTED/ACTA Press.)

6.3.2 Evaluation of the proposed pronunciation modeling method

To generate a multiple pronunciation dictionary, we performed a phoneme recognition and obtained a 200-best list for each utterance in the development set. As a phoneme recognizer, we used the baseline acoustic models, a phoneme based back-off bigram language model, and a pronunciation dictionary with a list of 41 phonemes with silence. By using the 200-best list, the performance of phoneme recognition was improved from 28.27% to 49.08%. In addition, the rule patterns could be generated using only phoneme sequences where the phoneme recognition accuracy was over 50%. After applying the rule patterns in C4.5, at a pruning option of 25%, we obtained 334 rules from the decision trees. Then, a multiple pronunciation dictionary was generated by adapting the baseline pronunciation dictionary from the obtained 334 rules. To reduce the confusability, we also applied the proposed optimization method to the adapted multiple pronunciation dictionary.

Table 1 compares the average WERs, the pronunciation dictionary size, and the ASR decoding time for the baseline ASR system and the ASR systems employing the multiple pronunciation dictionary and the pruned multiple pronunciation dictionaries according to pruning thresholds of 0, 0.1, 0.2, and 0.3. It can be seen in the table that the ASR system employing the multiple pronunciation dictionary increased the WER, compared to that employing the baseline pronunciation dictionary. The performance degradation incurred by the proposed multiple pronunciation dictionary was due to the increased confusability by improper pronunciation variants.

Next, the multiple pronunciation dictionary was pruned using the proposed confusability measure, and the average WERs of the ASR system using the differently pruned multiple pronunciation dictionaries are shown in the third row of Table 1. The table shows that the pruned multiple pronunciation dictionary constructed with a threshold of 0.1 gave the lowest average WER among all other dictionaries. That is, the average WERs of an ASR system using the pruned multiple pronunciation dictionary were 18.58% and 0.59% for non-native and native speech, respectively, which corresponded to relative WER reductions of 6.98% and 15.30%, compared to those of the baseline ASR system and an ASR system using the multiple pronunciation dictionary prior to pruning. Moreover, the ASR decoding time for the pruned multiple pronunciation dictionary was also reduced by 21.10% compared to that for the multiple pronunciation dictionary without pruning.

7. Conclusion

This chapter addressed issues associated with efficient pronunciation variation modeling for non-native automatic speech recognition (ASR), where non-native speech was mostly characterized by different pronunciations, speaking styles, and articulators of speakers from their native speech. The techniques for improving the performance of non-native ASR could then be classified into four approaches: acoustic modeling, language modeling, pronunciation modeling, and hybrid modeling approaches. We first reviewed these four approaches before proposing a new pronunciation model adaptation method.

In particular, the proposed pronunciation adaptation method was based on a multiple pronunciation dictionary, designed using an indirect data-driven method. However, this approach resulted in an increased search space for ASR decoding due to the increase of the pronunciation dictionary size. Therefore, a method for optimizing the size of the multiple pronunciation dictionary was also proposed, where a confusability measure based on the Levenshtein distance was introduced in order to remove some confusable pronunciation variants from the dictionary. To investigate the effects of the proposed approach on ASR performance, English was selected as the target language and English utterances spoken by Koreans were considered as the non-native speech. Subsequently, it was shown from the continuous non-native ASR experiments that an ASR system using the optimized multiple pronunciation dictionary could achieve an average word error rate reduction of 15.30%, with a relative reduction in computational complexity of 21.10%, compared to that achieved using the multiple pronunciation dictionary without optimization.

8. References

Alotaibi, Y. A. & Muhammad, G. (2010). Study on pharyngeal and uvular consonants in foreign accented Arabic for ASR, *Computer Speech and Language*, Vol. 24, No. 2, Apr. 2010, pp. 219-231.

Amdal, I.; Korkmazasky, F. & Suredan, A. C. (2000). Data-driven pronunciation modelling for non-native speakers using association strength between phones, *Proceedings of ISCA Tutorial and Research Workshop on Automatic Speech Recognition*, pp. 85-90, Paris, France, Sept. 2000.

Amdal, I.; Korkmazskiy, F. & Surendran, A. C. (2000). Joint pronunciation modelling of non-native speakers using data-driven methods, *Proceedings of 6th International Conference on Spoken Language Processing (ICSLP)*, pp. 622-625, Beijing, China, Oct. 2000.

Bartkova, K. & Jouvet, D. (2006). Using multilingual units for improved modeling of pronunciation variants, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1037-1040, Toulouse, France, May 2006.

Bellegarda, J. (2001). An overview of statistical language model adaptation, *Proceedings of ITRW on Adaptation Methods for Speech Recognition*, pp. 165-174, Sophia Antipolis, France, Aug. 2001.

Bouselmi, G.; Fohr, D.; Illina, I. & Haton, J. P. (2006). Fully automated non-native speech recognition using confusion-based acoustic model integration and graphemic constraints, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 345-348, Toulouse, France, May 2006.

Bouselmi, G.; Fohr, D. & Illina, I. (2007). Combined acoustic and pronunciation modelling for non-native speech recognition, *Proceedings of 8th Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 1449-1452, Antwerp, Belgium, Aug. 2007.

Compernolle, D. V. (2001). Recognizing speech of goats, wolves, sheep and ... non-natives, *Speech Communication*, Vol. 35, Nos. 1-2, Aug. 2001, pp. 71-79.

Fosler-Lussier, E. (1999). Multi-level decision trees for static and dynamic pronunciation models, *Proceedings of 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 463-466, Budapest, Hungary, Sept. 1999.

Goronzy, S.; Sahakyan, M. & Wokurek, W. (2001). Is non-native pronunciation modelling necessary?, *Proceedings of 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 309-312, Aalborg, Denmark, Sept. 2001.

Goronzy, S.; Rapp, S. & Kompe, R. (2004). Generating non-native pronunciation variants for lexicon adaptation, *Speech Communication*, Vol. 42, No. 1, Jan. 2004, pp. 109-123.

Gruhn, R.; Markov, K. & Nakamura, S. (2004). A statistical lexicon for non-native speech recognition, *Proceedings of 8th International Conference on Spoken Language Processing (ICSLP)*, pp. 1497-1500, Jeju Island, Korea, Oct. 2004.

He, X. & Zhao, Y. (2003). Fast model selection based speaker adaptation for nonnative speech, *IEEE Transactions on Speech and Audio Processing*, Vol. 11, No. 4, July 2003, pp. 298-307.

Hernandez-Abrego, G.; Olorenshaw, L.; Tato, R. & Schaaf, T. (2004). Dictionary refinements based on phonetic consensus and non-uniform pronunciation reduction, *Proceedings of 8th International Conference on Spoken Language Processing (ICSLP)*, pp. 1697-1700, Jeju Island, Korea, Oct. 2004.

Huang, C.-L.; Wu, C.-H.; Li, H.; Hsieh, C.-H. & Ma, B. (2008). Unsupervised pronunciation grammar growing using knowledge-based and data-driven approaches, *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1097-1100, Hannover, Germany, June 2008.

International Phonetic Association. (2008). International Phonetic Association. *Handbook of the International Phonetic Alphabet: A Guide to the Use of the International Phonetic Alphabet*, Cambridge, UK.

Kim, M.; Oh, Y. R. & Kim, H. K. (2007). Non-native pronunciation variation modeling using an indirect data-driven method, *Proceedings of 10th biannual IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 231-236, Kyoto, Japan, Dec. 2007.

Kim, M.; Oh, Y. R. & Kim, H. K. (2008). Optimizing multiple pronunciation dictionary based on a confusability measure for non-native speech recognition, *Proceedings of Artificial Intelligence and Applications (AIA 2008)*, pp. 215-220, Innsbruck, Austria, Feb. 2008.

Levenshtein, V. I. (1996). Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics Doklady*, Vol. 10, No. 8, Feb. 1996, pp. 707-710.

Liu, L.; Zheng, T. F. & Wu, W. (2008). State-dependent phoneme-based model merging for dialectal Chinese speech recognition, *Speech Communication*, Vol. 50, No. 7, July 2008, pp. 605-615.

Matsunaga, S.; Ogawa, A.; Yamaguchi, Y. & Imamura, A. (2003). Non-native English speech recognition using bilingual English lexicon and acoustic models, *Proceedings of IEEE*

International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 340-343, Hong Kong, China, Apr. 2003.

Morgan, J. (2004). Making a speech recognizer tolerate non-native speech through Gaussian mixture merging, *Proceedings of InSTIL/ICALL Symposium on Computer Assisted Learning*, paper 052, Venice, Italy, June 2004.

Oh, Y. R.; Yoon, J. S. & Kim, H. K. (2007). Acoustic model adaptation based on pronunciation variability analysis for non-native speech recognition, *Speech Communication*, Vol. 49, No. 1, Jan. 2007, pp. 59-70.

Oh, Y. R.; Kim, M. & Kim, H. K. (2008). Acoustic and pronunciation model adaptation for context-independent and context-dependent pronunciation variability of non-native speech, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4281-4284, Las Vegas, NV, Apr. 2008.

Oh, Y. R. & Kim, H. K. (2009). MLLR/MAP adaptation using pronunciation variation for non-native speech recognition, *Proceedings of 11th biannual IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 216-221, Merano, Italy, Dec. 2009.

Oh, Y. R. & Kim, H. K. (2010). On the use of feature-space MLLR adaptation for non-native speech recognition, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4314-4317, Dallas, TX, Mar. 2010.

Paul, D. & Baker, J. (1992). The design for the Wall Street Journal-based CSR corpus, *Proceedings of 5th DARPA Speech and Natural Language Workshop*, pp. 357-362, Harriman, NY, Feb. 1992.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA.

Raab, M.; Gruhn, R. & Noeth, E. (2007). Non-native speech databases, *Proceedings of 10th biannual IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 413-418, Kyoto, Japan, Dec. 2007.

Raux, A. (2004). Automated lexical adaptation and speaker clustering based on pronunciation habits for non-native speech recognition, *Proceedings of 8th International Conference on Spoken Language Processing (ICSLP)*, pp. 613-616, Jeju Island, Korea, Oct. 2004.

Raux, A. & Eskenazi, M. (2004). Using task-oriented spoken dialogue systems for language learning: Potential, practical applications and challenges, *Proceedings of InSTIL/ICALL Symposium on Computer Assisted Learning*, paper 035, Venice, Italy, June 2004.

Rhee, S.-C.; Lee, S.-H.; Kang, S.-K. & Lee. Y.-J. (2004). Design and construction of Korean-spoken English corpus, *Proceedings of 8th International Conference on Spoken Language Processing (ICSLP)*, pp. 2769-2772, Jeju Island, Korea, Oct. 2004.

Schaden, S. (2003). Generating non-native pronunciation lexicons by phonological rules, *Proceedings of International Congress of Phonetics Sciences*, pp. 2545-2548, Barcelona, Spain, Aug. 2003.

Sidaras, S. K.; Alexander, J. E. D. & Nygaard, L. C. (2009). Perceptual learning of systematic variation in Spanish-accented speech, *Journal of the Acoustical Society of America*, Vol. 125, No. 5. May 2009, pp. 3306-3316.

Strik, H. & Cucchiarini, C. (1999). Modeling pronunciation variation for ASR: A survey of the literature, *Speech Communication*, Vol. 29, Nos. 2-4, Nov. 1999, pp. 225-246.

Steidl, S.; Stemmer, G.; Hacker, C. & Noth, E. (2004). Adaptation in the pronunciation space for non-native speech recognition, *Proceedings of 8th International Conference on Spoken Language Processing (ICSLP)*, pp. 2901-2904, Jeju Island, Korea, Oct. 2004.

Svendsen, T. (2004). Pronunciation modeling for speech technology, *Proceedings of International Conference on Signal Processing and Communications (SPCOM)*, pp. 11-16, Bangalore, India, Dec. 2004.

Tajchman, G.; Fosler, E. & Jurafsdy, D. (1995). Building multiple pronunciation models for novel words using exploratory computational phonology, *Proceedings of 4th European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 2247-2250, Madrid, Spain, Sept. 1995.

Tan, T. & Besacier, L. (2007). Acoustic model interpolation for non-native speech recognition, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1009-1012, Honolulu, HA, Apr. 2007.

Tsai, M.; Chou, F. & Lee, L. (2002). Improved pronunciation modeling by properly integrating better approaches for baseform generation, ranking and pruning, *Proceedings of ISCA Tutorial and Research Workshop (ITRW) on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology*, pp. 77-82, Estes Park, CO, Sept. 2002.

Wade, T.; Jongman, A. & Sereno, J. (2007). Effects of acoustic variability in the perceptual learning of non-native-accented speech sounds, *Phonetica*, Vol. 64, Nos. 2-3, Aug. 2007, pp. 122-144.

Wang, Z.; Schultz, T. & Waibel, A. (2003). Comparison of acoustic model adaptation techniques on non-native speech, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 540-543, Hong Kong, China, Apr. 2003.

Weide, H. (1998). *The CMU Pronunciation Dictionary, release 0.6*, Carnegie Mellon University, Pittsburgh, PA.

Wiseman, R. & Downey, S. (1998). Dynamic and static improvements to lexical baseforms, *Proceedings of ESCA Tutorial and Research Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, pp. 157-162, Rolduc, The Netherlands, May 1998.

Wolff, M.; Eichner, M. & Hoffmann, R. (2001). Automatic learning and optimization of pronunciation dictionaries, *Proceedings of ITRW on Adaptation Methods for Speech Recognition*, pp. 159-162, Sophia Antipolis, France, Aug. 2001.

Yang, J.; Pu, Y.; Wei, H. & Zhao, Z. (2004). Acoustic models adaptation in large vocabulary continuous Mandarin speech recognition for non-native speakers, *Proceedings of International Conference on Signal Processing (ICSP)*, pp. 687-690, Beijing, China, Sept. 2004.

Young, S.; Odell, J. & Woodland, P. (1994). Tree-based state tying for high accuracy acoustic modelling, *Proceedings of ARPA Human Language Technology Workshop*, pp. 307-312, Plainsboro, NJ, Mar. 1994.

Young, S.; Evermann, G.; Kershaw, D.; Moore, G.; Odell, J.; Ollason, D.; Povey, D.; Valtchev, V. & Woodland, P. (2002). *The HTK Book (for HTK Version 3.2)*, Cambridge, UK.

7

Applications of Speech Technologies in Western Balkan Countries

Darko Pekar[1], Dragiša Mišković[1,2], Dragan Knežević[1,2], Nataša Vujnović Sedlar[1,2], Milan Sečujski[2] and Vlado Delić[2]
[1]*AlfaNum – Speech Technologies, Novi Sad,*
[2]*Faculty of Technical Sciences, University of Novi Sad, Serbia*

1. Introduction

The chapter will present the first applications of speech technologies in the countries of Western Balkans, launched by the Serbian company AlfaNum. The speech technologies for Serbian and kindred South Slavic languages are developed in cooperation with the University of Novi Sad, Serbia. Most of these applications are rather innovative in Western Balkans and they will serve as a base for complex systems which will enable 20 millions of inhabitants of this part of Europe to talk to machines in their midst in their native languages, equally to their counterparts who live in more developed countries in the region.
Firstly, the importance of research and development of speech technologies will be stressed, particularly in view of their language dependence and, on the other hand, the possibility of their wide application. The central part of the chapter will focus on the results of the research and development of the first applications of automatic speech recognition (ASR) and text-to-speech synthesis (TTS) across Western Balkans – some of them are a novelty in a much wider region as well. The paper will be concluded by the directions of future research and development of new applications of speech technologies in the Western Balkan region and worldwide.

1.1 Relevance of the research and development of speech technologies

When communicating with others, people predominantly use the senses of sight and hearing – they speak, listen and watch. On the other hand, when communicating with machines (computers, telephones, robots, cars etc.), they mostly use the senses of sight and touch – they look at monitors and touch keyboards, mice or touch screen displays. It is worth noting that humans rarely address machines using speech and that machines rarely use speech to respond, although spoken communication is the most natural form of communication among humans. Apart from a number of fundamental problems related to ASR and TTS applications, addressed in more detail in (Delić et al., 2010), another possible reason for this is the fact that speech technologies are highly language dependent, and that a number of necessary resources and techniques have to be developed for each language separately. The most has been done for languages spoken by relatively large communities, but quality solutions for languages with smaller communities are beginning to emerge.

ASR is language dependent to a great extent, and TTS to an even greater. There are several aspects of this language dependence: (1) A database of at least several hours of recorded speech in a specific language must exist, in order to be able to produce high-quality synthesised speech, regardless of the method used. Speech databases for ASR training, which can be of much greater size, are also language dependent. (2) Morpho-syntactic analysis and syntactic-prosodic parsing of the input text have to be carried out, and both tasks are highly language dependent. (3) Based on the previous analysis of input text, appropriate prosody features (phone duration, f_0 contour and energy) have to be generated.

1.2 Integration of ASR and TTS engines into applications

AlfaNum TTS and ASR engines can be used through a number of interfaces, all of them built upon basic TTS and ASR libraries written in C++. The main reason for their design was to make engine integration into existing products as simple and as fast as possible.

- **C++ library (proprietary interface)** – TTS and ASR C++ libraries are at the base of all supported interfaces.
- **Microsoft SAPI** – The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. Both SAPI 4 and SAPI 5.x interfaces are implemented. In general, all versions of the API have been designed such that a software developer can write an application to perform speech recognition and synthesis by using a standard set of interfaces, accessible from a variety of programming languages. Speech engines are built as standard COM objects by implementing the required COM interfaces.
- **Media Resource Control Protocol** – The Media Resource Control Protocol (MRCP) is a protocol proposed by the Internet Engineering Task Force (IETF), which has the goal of standardising computer dialogues between the ASR and TTS with interactive voice response (IVR). Clients send MRCP messages to the server over a network usually by means of another protocol, such as Real Time Streaming Protocol (version 1) or Session Initiation Protocol (version 2). AlfaNum servers comply with version 2 of MRCP protocol.
- **AlfaNum IP server/client (proprietary interface)** – This interface is based on a proprietary protocol which includes additional functionality not found in any of the industry standard protocols. This protocol is designed to make the system more robust and provides faster content delivery. For this purpose speech engines (C++ libraries) are built into the AlfaNum IP servers. Along with the server, TTS and ASR client libraries are created to enable developers the use of AlfaNum IP server functionality from within different programming languages. Client libraries are developed for C++, C#, Visual Basic and PHP programming languages.

  The basis of AlfaNum IP server is a multi-threading protocol which accepts connections from client applications and is based on TCP/IP. The functioning of the server can be explained through two types of sockets that are created. The first one is the *listening socket*, which collects connection demands generated by clients. After the demands are received, a *service socket* is opened for each client, through which further communication is carried out, as shown in Fig. 1. Such a mechanism enables handling a large number of users and simple addition of new routines.

  Besides remote access, the client library that encompasses the communication between the applications and the server also enables the use of multiple ASR/TTS servers (located at different computers) in case of need for a large numbers of simultaneous requests for speech recognition.

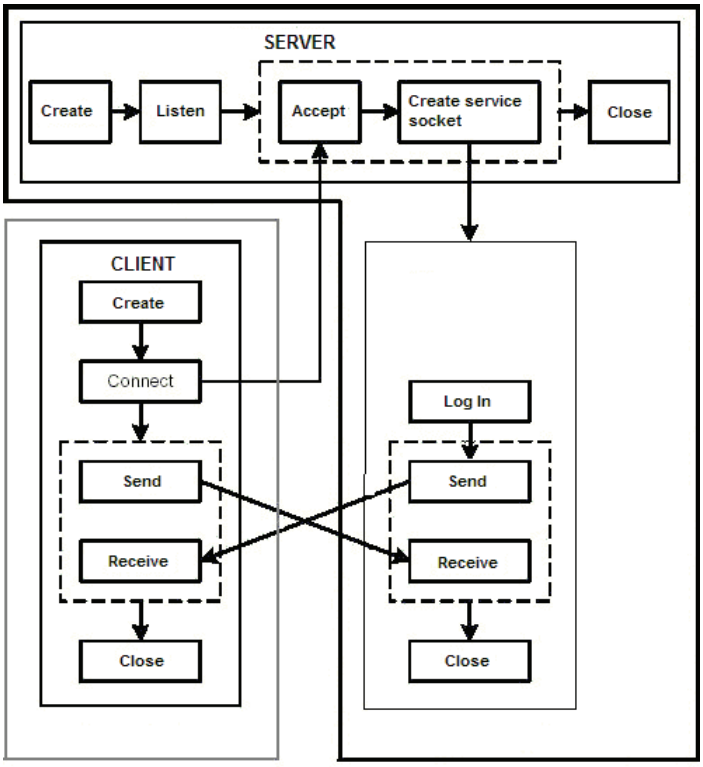


Fig. 1. Communication protocol of the AlfaNum IP server

Within speech-enabled applications ASR components are commonly coupled with TTS components, thus completing the cycle of human-machine speech communication (Delgado & Araki, 2005). Various areas of interaction have so far been covered, and today there are active systems offering e.g. information related to bus schedules and stock market, as well as any information that can commonly be found in electronic newspapers.

2. ASR applications

The public telephone network is currently the most promising ground for application of speech technologies (Nöth, 2004). The first applications of ASR in Western Balkans have been launched at the public telephone network, with support by intelligent network functionalities. Some of the innovative applications of ASR in Serbia will be described in the following sections.

2.1 Interactive Voice Response systems

As has been mentioned before, ASR and TTS IP servers have found their first applications within AlfaNum Interactive Voice Response (IVR) systems. An IVR system is a computerised system allowing a user (a telephone caller) to choose among various options offered in voice menus. The first IVR systems played pre-recorded voice prompts to which the user

would press a number on a telephone keypad to select the option. Integration of ASR and TTS components significantly improve this interaction and complete the cycle of human-machine communication. The foundation of all IP server based IVR systems developed is the simultaneous functionality of ASR and TTS servers and their communication with a required number of IVR processes (one per telephone line) via IP protocol.

Intel/Dialogic Telephony Cards provide a connection to the public telephone network. Through it, the calls are routed to any of the free channels managed by the IVR controller. At the same time, the controller provides a link to the database and ASR and TTS servers. The database represents an information source from which data is presented to the user by TTS in the form of synthesised speech, based on user requests that the system acquires via ASR. The ASR and TTS servers can reside on remote computers (dedicated if required) and can communicate with a number of different IVR applications.

Specific properties of such systems, from the point of view of ASR, manifest themselves in the need for activation of different recognisers according to options offered to the user in a given moment. Furthermore, the systems handle information that changes dynamically, and for that reason the grammars used for recognition often have to be generated dynamically according to the database contents. The basic organisation of an AlfaNum IVR system is shown in Fig. 2.

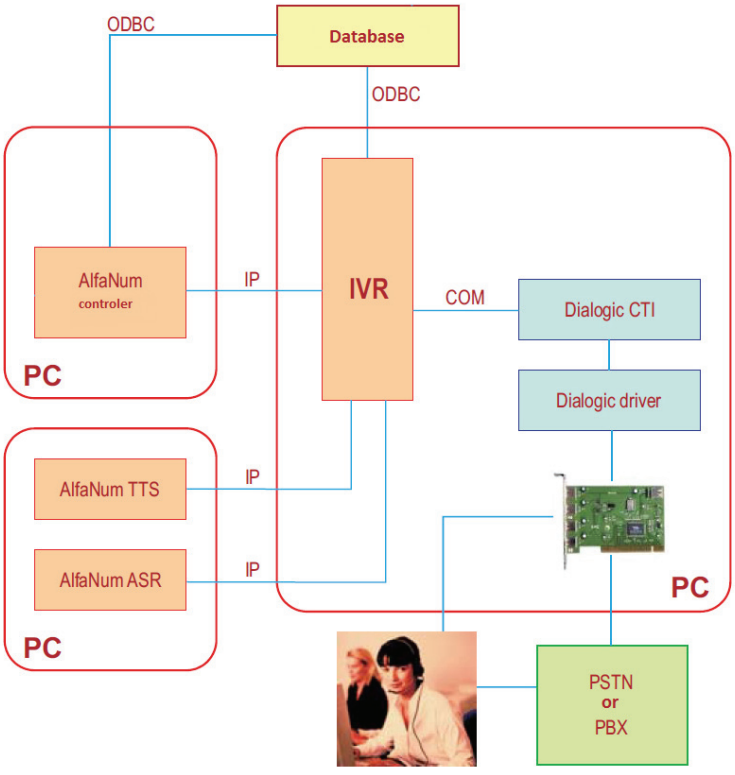


Fig. 2. The basic organisation of an AlfaNum IVR system

2.2 Advertising monitor

Besides application in the field of telephony, the AlfaNum ASR engine has also been implemented in systems that perform searches through a large amount of audio material. One of such systems, Advertising Monitor (Pekar et al., 2007), is an application that locates audio content such as jingles and commercials in audio archives. The system comprises a number of FM and TV tuners receiving signals from various radio and TV stations, and a search service that can be distributed to multiple computers. Specific properties of the material being searched allow the use of a simplified recognition process based on LPC coefficients. Unlike classical speech recognition, the input to this system is subject to different types of variations, which is reflected in the sound signal processing algorithm. However, there are also some alleviating circumstances for development of such a system:

- A complete absence of temporal variability between the reference recording and the test recording.
- A drastical reduction in acoustic variability in comparison with classical speech recognition. In this case, acoustic variability is the consequence of changes in channel properties (spectral changes and noise), which evolve slowly over time and the effect of which can be reduced to a sufficient degree using first time derivatives of acoustic features.

For that reason, processing of the sound signal amounts to calculation of its dynamic features, namely, first and second time derivatives of LPC coefficients. In this way there is no front-end processing in recognition and a significant portion of the processor time can be saved. Furthermore, because of the aforementioned absence of temporal variability, simple one-on-one comparison of the reference recording and the incoming signal can be applied instead of DTW or another, more complex time-alignment algorithm. Blocks containing the reference recording simply slide along the received signal and block-by-block comparison is carried out through calculation of the average distance between blocks of the reference recording and corresponding points in the received signal. When a very significant drop in the distance is observed, it can be concluded that the reference recording was located in the received signal, as shown in Fig. 3.

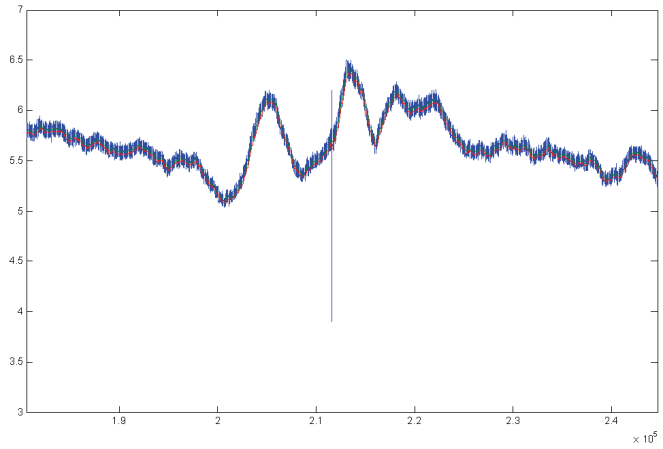


Fig. 3. Distribution of the average distance between the reference recording and the received signal

The system can fail only in cases where there is a significant mismatch between the original reference recording and the occurrence of the same audio material in the received signal (e.g. a truncated or overdubbed commercial). However, such problems can be efficiently handled by appropriate postprocessing techniques.

2.3 Word Spotter

AlfaNum Word Spotter (Mišković et al., 2007) is an application that locates key words and phrases, given as text, in arbitrary audio material. The system relies on the ASR speech recognition engine, and the nature of the system indicates its areas of use and features expected by its users (various security agencies, media monitoring agencies etc.).

The functioning of the AlfaNum Word Spotter is based on the phoneme-based, speaker independent speech recognition system, AlfaNum ASR. Particular features of the application are related to the way trellises for given key words and phrases are built. If the standard approach to speech recognition is taken, with adaptation of syntax so as to allow for multiple pronunciations of a single word, word spotter produces the recognition result as a sequence of arbitrary number of silence models, noise ("garbage") models, key word models and wildcard models (universal models covering parts of an utterance that do not contain key words). This is the consequence of the way the trellis for each key word is generated, as shown in Fig. 4 (word 1,... word *n* represent transcriptions of basic and inflected forms of a word).

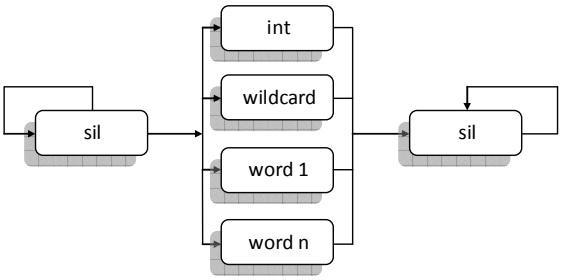


Fig. 4. Transition diagram of the AlfaNum Word Spotter

As can be seen in the figure, besides key words and phrases, trellis also contains non-speech states as well as states modelling various types of noise (INT). Having such a structure in mind, buildup of a trellis of the word spotter is based on the following rules: each sequence must begin with silence (non-speech state of an unlimited duration), from silence it is possible to traverse only into an initial state of a key word or phrase or into a noise or wildcard model, a word has linear structure and limited duration, from the final state of a key word or phrase it is possible to traverse only into the state of silence, which is at the end of every sequence.

Unlike application of ASR in interactive voice response systems or call centres, where a user can be asked to repeat the utterance more clearly in case of unreliable recognition, in case of a word spotter error rate has to be reduced to a minimum possible level.

Two types of errors can be identified. The first type of errors is related to key words that existed in the recording, but were not recognised by the system (false negatives), and such errors are critical from the point of view of system reliability. The second type of errors is related to the words that did not exist in the recording, but were nevertheless "recognised"

by the system (false positives). Eliminating as much false positives as possible without creating significant false-negative results is a very demanding task, directly related to specific properties of ASR algorithms. Some of the false positives can be eliminated by subsequent comparison of the durations of particular phonetic segments of the recognised word or phrase to the expected ones (Mišković at al., 2007). The graphical user interface of the application has been designed so as to enable the user to eliminate a significant number of false positives, since the recognition results (recognition locations) are displayed in order of decreasing reliability. The user can thus decide to stop manually checking the results when a sufficiently high rate of false positives is reached. A portion of the graphical user interface related to recognition result verification is shown in Fig. 5. The figure shows the situation after some of the results have been checked, and the distribution of accurate recognitions vs. false positives can be observed.

The next step in the development of this tool would be in the direction of its integration with a system for recording telephone conversations.

Besides the applications described in this section, developed on the basis of the existing system for speech recognition, there is a number of areas in which the application of this system is yet to be expected. Ongoing development of a continuous speech recognition

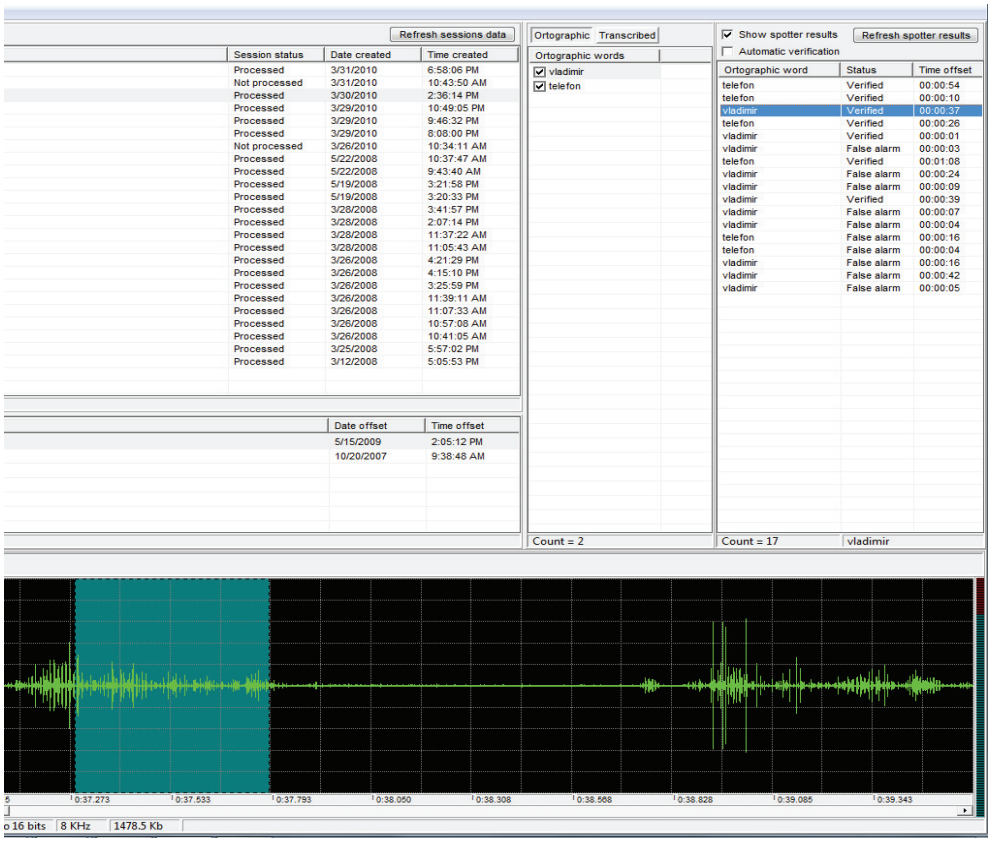


Fig. 5. A segment of the graphical user interface of the AlfaNum Word Spotter

system for large vocabularies expands the area of application of this technology. The first applications are expected to be dictation systems, spoken dialogue systems and applications for automatic subtitling of TV shows. Applications related to speaker identification and verification will also be developed.

3. TTS applications

This section will describe the first applications of text-to-speech synthesis in Serbian and kindred South Slavic languages. The AlfaNum TTS engine supports a number of interfaces in order to facilitate its integration into useful applications. Some of these interfaces are standard, such as the C++ library, Microsoft SAPI5, Microsoft SAPI4, and MRCP, however, communication with other components in sophisticated systems is also possible via a custom designed AlfaNum IP protocol (implemented through libraries in C++, Visual Basic, C# and PHP).

3.1 AnReader

The first widely applied TTS-based system in WBCs is *anReader* (Delić et al., 2005; Sečujski et al., 2007), used by almost one thousand visually impaired persons in Serbia, Bosnia and Herzegovina, Montenegro, Croatia and FYR Macedonia. Before the appearance of *anReader*, the most widely used system was *WinTalkerVoice*, originally built for Czech language. It produced synthesised speech of poor quality in Serbian and Croatian, and has therefore never been used for any other purpose than as aid for the visually impaired. *AnReader*, on the other hand, was initially developed for Serbian, and later for Croatian and Macedonian as well. The basic concepts are the same, but the morphological dictionary (especially the information related to accentuation) and the rules for morpho-syntactic analysis had to be modified. The Croatian *anReader* required that a new speech database in Croatian be recorded and processed, while the Macedonian version currently uses the Serbian speech database, with slight impairment of speech quality as a result. High-level speech synthesis of Serbian and Croatian is performed using expert POS taggers, while for Macedonian full POS-tagging is never performed since it is not necessary for reasonably natural pronunciation of Macedonian (owing to the simplicity of accentuation in Macedonian in comparison to the other two languages).

It should be kept in mind that, for a visually impaired user to be able to use a computer unaided, besides a synthesiser such as *anReader*, he/she also needs a screen-reader, an application attempting to identify and interpret what is being displayed on the screen, as well as to communicate information on menus, controls, and other visual constructs. Owing to a number of freely available screen-readers, a quality speech synthesiser remains the critical component needed by any visually impaired individual for unaided computer access.

Owing to its superiority, *anReader* has quickly gained popularity among the visually impaired computer users in all of the countries of Western Balkans, and its use has resulted in a tenfold multiplication in their number, earning it the status of an official aid for the visually impaired, available to the visually impaired in Serbia through the Institute for Health and Social Care of the Republic of Serbia.

The new, higher quality of synthesised speech in Serbian and the potentials of its TTS engine for South Slavic languages were recognised very soon and, consequently, *anReader* was awarded the first prize of the Serbian Society of Informatics as the best applied software product in 2004.

3.2 Audio library for the visually impaired

There are more than 10.000 persons in Serbia with a visual disability of some kind, and a much larger number throughout the region of Western Balkans. The greatest centre for education of the visually impaired in Serbia and the entire Western Balkans is the School for the Visually Impaired Children „Veljko Ramadanović" in Zemun. Until the introduction of the Audio library for the visually impaired (ABSS) (Mišković et al., 2005), written information necessary for education of the pupils of this school had been available in the form of Braille books, which are well known to be very impractical and extremely expensive to prepare, store and maintain, as well as audio recordings of books read out by human speakers, which have basically the same drawbacks. Preparation of both Braille and audio-books is also a lengthy process, making them quite inconvenient as media for accessing constantly changing information.

The Audio library for the visually impaired was developed in answer to these problems. It is a web-accessible client-server system in which a large quantity of books and texts from other sources is stored at the server side, while the client application enables an individual user to access the desired text, download it and have it converted to speech using a TTS system, namely, *anReader*. Searches by author name, genre and content are supported, and navigation through texts is intuitive and efficient due to a number of useful options. The texts are stored in an encrypted format in order to ensure the legal rights of copyright owners, preventing the users from being able to copy or print them.

As mentioned before, the Audio library is organised as a client-server application (Fig. 6). The administrator application, the database of books and the server in charge of handling user requests and accessing the database are situated on the server side. The books are internally stored in HTML format, which facilitates the retrieval of particular paragraphs before actual synthesis of speech. The client side contains an application intended for direct interaction with the visually impaired users as well as network communication. The user interface comprises two modules – *anAdministrator* (on the server side) and *anKlijent* (on the client side). The *anAdministrator* module is in charge of library administration, enabling inclusion and management of new books as well as search for (and within) the existing. The latest version of the library (Mišković et al., 2006) is multilingual, taking full advantage from

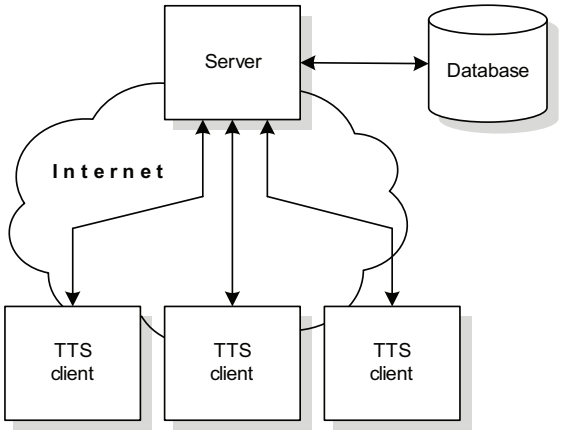


Fig. 6. Internal organisation of the Audio library

the fact that *anReader* has been developed for Croatian and Macedonian language as well, and that speech synthesis integrated into MS Windows can be used for reading books in English as well.
The *anAdministrator* module is completely speech enabled, which in turn enables the visually impaired to administrate the library themselves. The functionality of the application (connecting to the database and performing queries) is realised through ODBC (Open Database Connectivity) drivers for MySQL. The *anKlijent* module is speech enabled as well, which eliminates the need for any additional screen reader (a solution that would be hardly possible to use anyway, since the complete interface is in the language of the user's choice).
The executive module of the entire application is *anKlijent*, which relies on the communication module and on Microsoft SAPI 5.0, which provides access to Windows virtual speakers. The SAPI interface offers a number of advantages related to automatic handling of audio-devices, multi-threading, speaker selection etc. For that reason, besides AlfaNum TTS, which is implemented as two virtual speakers, *anKlijent* can also use the original Windows speech synthesis, which is suitable for handling texts written in English.
The initial version of the Audio library used RPC (*Remote Communication Protocol*) for communication. However, introduction of web access in the version 2.0 required implementation of new routines and a higher degree of control. For that reason, it was necessary to implement a custom protocol, based on ASR and TTS IP servers, which better answered the needs of dial-up users in particular, and the AlfaNum IP server, described in detail in section 1.2, was used for this purpose. Thus, the library has become a system independent from the actual location of the server and the database of books, since it allows the use of Internet for client-server communication.
The Audio library, as such, represents a significant step towards equality in education and access to information for the visually impaired. It is also a very convenient tool for all those who prefer textual content to be read out to them aloud while they are busy performing other tasks at their computers.

3.3 Voice enabled web sites

One of the recently developed applications of speech synthesis is enabling arbitrary web sites with speech synthesis through an IP TTS server particularly designed for this purpose. Owing to this system, visitors of web sites are able to listen to textual content instead of reading it, leaving their eyes free for some other task.
The interface to the server is remarkably simple, based on a PHP library and an accompanying javascript, facilitating integration of TTS functionality into existing web sites with minimal human intervention. The PHP library is universal, and the javascript is easily adaptable to each particular website.
The TTS server is optimised for enabling websites with speech synthesis through streaming of mp3 compressed sound, which results in virtually instantaneous server response. The server contains a minimum HTTP server within, which supports GET requests for file delivery by responding to them by direct sending of mp3 streams from the buffer in case the entire text has not yet been synthesised or by sending the recorded file in case the synthesis has been accomplished. Mp3 content is delivered to a Flash mp3 player embedded into the client application.
The process of requesting and obtaining synthesised speech can be summarised as follows (Fig. 7):

- WEB browser issues a request for a speech enabled web page;
- WEB server issues a TTS request;
- TTS server initiates synthesis and responds by sending the synthesised file name back to the WEB server;
- WEB server responds to the WEB browser by sending HTML content (with the embedded player's "file" parameter set to the actual name of the synthesised file);
- WEB browser displays the page and requests mp3 encoded speech from the TTS server;
- TTS server's embedded HTTP server responds by sending the mp3 stream.

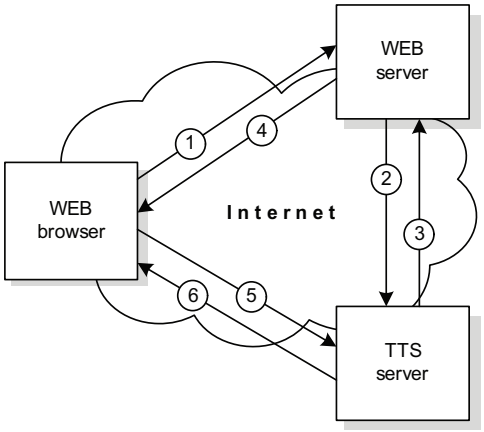


Fig. 7. Retrieval of synthesised speech from a speech enabled web site

The first speech enabled web site in the Western Balkan region is the site of Radio television Vojvodina (the northernmost province of Serbia) (http://www.rtv.rs), using the AlfaNum TTS engine. This web site was speech enabled in May 2009. After this pilot project has been successfully carried out, the interest for this web site feature across the Western Balkan countries has been on the increase and the AlfaNum team has recently obtained support from the Ministry of Science and Technological Development of the Republic of Serbia in the effort of enabling a significant number of Serbian web sites with speech.

4. ASR&TTS applications

4.1 Web portal Kontakt

Text-to-speech, as a technology with a wide range of application, becomes even more powerful when coupled with ASR. One of the examples of this is the web portal Kontakt, developed by the same team (Ronto & Pekar, 2005). This portal, intended primarily for visually impaired and elderly users, can be accessed by both computer users and those who do not own or use a computer since it is accessible by telephone as well, and it is essentially an Internet site whose contents are updated automatically from the websites of 4 well-known news sites in Serbia. Furthermore, authorised users can access it and submit information of particular interest to the visually impaired and/or the elderly. Each time the contents of the website are updated, the menu structure in the interactive voice response (IVR) interface is updated automatically as well. The users can, thus, navigate the site

through voice commands and receive information via synthesised speech. Through the same interface, the users can change the speaker, speech rate and pitch, according to their own preferences. The portal is accessible via the intelligent network at a 0700 telephone number, which means that each user pays only the price of the local telephone call, regardless of the actual origin of the call.

The portal relies on a speech database as a source of information. In order to present the requested information to the user, it is necessary to send a query to the database, and receive the requested information in response. For the system to be efficient enough, it has to provide simultaneous access to information to a sufficient number of users. If the portal is accessed via telephone, the entire human-machine communication is carried out via speech.

In this case, as presented in Fig. 8, the communication in the system is based on interactive voice response (IVR) applications which handle one telephone line each through ASR and TTS IP servers and retrieve the requested information from the database mentioned above. The advantage of such a solution is in the fact that ASR and TTS servers can be remote and dedicated exclusively to speech recognition and synthesis. A possible disadvantage would be the delay in response in case of server overload.

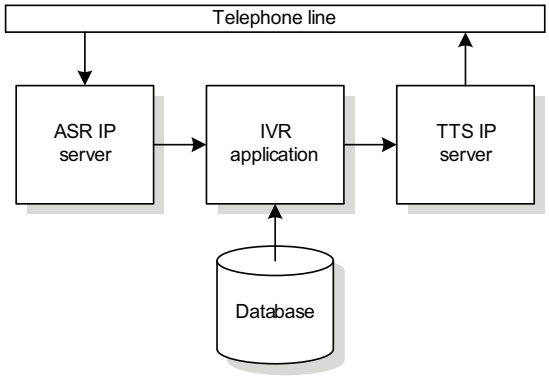


Fig. 8. Handling user requests in the system

The database in Fig. 8 is actually a MySQL database, which enables automated generation of web pages using PHP scripts, as well as a reliable connection with the IVR application realised through the C API of MySQL. The contents are refreshed using a puller application designed so as to gather new contents from the Internet on a periodical basis. These contents can be managed via an ordinary browser under administrative credentials. The system communicates with the telephone line using a Dialogic CTI card, and communication is controlled through Dialogic dx and Global Call API. The number of telephone lines that can be handled depends on the number and type of Dialogic cards integrated into the PC platform of the system.

Having in mind the technological limitations of ASR, users can form their queries in a standard format described by a number of *grammars*, which define the set of words, phrases and their combinations expected as input to the ASR process at any given moment. An example would be the initial grammar of the system, actual at the moment when the user initially addresses the system:

```
cmd = TEME | NASLOVI;[1]
gr = <gar>;
main = [$gr] [$cmd] [$gr];
```

where <gar> stands for any noise that is to be ignored during recognition. A successful navigation through a menu structure requires a new grammar to be defined at each point in the dialogue. The menu structure depends on the defined topics in the database, and on the other side, changes in the content of the database must have as little influence as possible on the design of the entire system. The only acceptable solution is to automatically generate grammars from the database. In the latest version of the system, grammars are defined at the initialisation of the IVR application. The ASR server is started thereafter, and thus it uses up-to-date grammars. The only deficiency of such an approach is that, if the database is refreshed while the application is active, the ASR server needs to be restarted.
For any newly generated grammar to be successfully used by the ASR server, it is necessary to communicate the location of each new grammar file to the server. This is done via the initialisation file of the ASR server, which contains all settings relevant to the functioning of the server, such as the parameters related to speech signal processing, recognition itself, IP port through which server communicates and other data related to the server. These data contain the vector of recognisers, defining the name, grammar file paths, postprocessor, pronunciation dictionary and phonetic transcriptor for each recogniser. In this context, the term "recogniser" denotes a set of rules to be used for recognition at a given moment. For the ASR server to be initialised with all newly generated grammar, it is necessary to establish a recogniser for each one of them, with all the necessary parameters, and include it into the vector of recognisers. From the point of view of the IVR application, defining all parameters of a recognition amounts to the selection of the appropriate recogniser.
The parameters of synthesis, on the other hand, can be configured by the users themselves. Each time a user logs out, the synthesis parameters of his/her choice are stored in the database, and the next time the user logs in, the same values are restored.
As such, the system was designed as a point of support to a number of the visually impaired and the elderly. As a project of great importance, the portal Kontakt has received support from Telekom Srbija, the Lottery of Serbia, as well as the community of the visually impaired in Serbia. The similar portal has been established in Croatia, with the only difference in that, at the moment, it updates its contents from a single news website.

4.2 iTEMA E-mail reader

iTEMA (Intelligent Telephone E-mail Access) is a multilingual CTI application for voice-enabled telephone access to user e-mails, developed within the joint EUREKA project E!3864 (Žganec Gros et al., 2006; Žganec Gros et al., 2008).
The architecture of the iTEMA system contains an interface towards a number of SAPI compatible TTS engines. The central element of the system is a dialogue manager connected to both telephone and Web interface (Fig. 9). Personal settings for each user, such as mobile phone number, PIN, e-mail access parameters, are stored in a database.
A user dials the number of the iTEMA user service and a human-machine dialog is initiated. Authentication is performed based on ANI and PIN, and followed by a personalised dialog enabling simple and intuitive navigation through a menu system. Through this dialog users

[1] which can be translated from Serbian as TOPICS | TITLES

can select messages they want to listen to, delete, or reply to using one of the pre-defined templates.
Beside drivers and business people, iTEMA also provides e-mail service to those who have difficulties when using a computer but use a telephone as a matter of routine (the visually impaired, many of the elderly etc.). The iTEMA project thus represents material support to the e-inclusion programme of the EU.

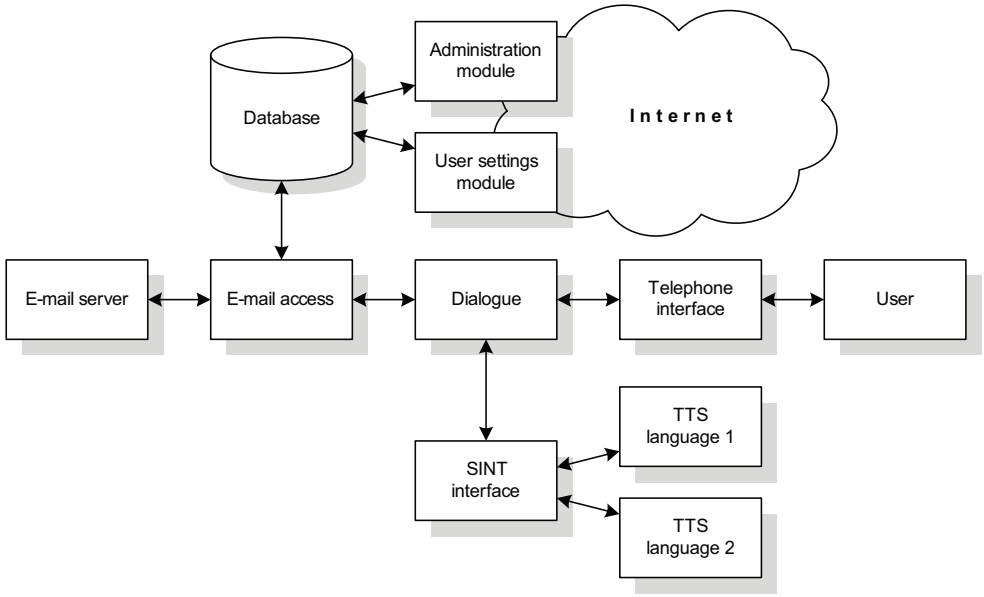


Fig. 9. Internal architecture of the iTEMA system

4.3 Computer games for the visually impaired

Besides the applications mentioned in the previous sections, the AlfaNum TTS engine, coupled with the AlfaNum ASR engine, was also used to create new computer games designed for entertainment and education of visually impaired children (Delić & Vujnović Sedlar, 2010; Lučić et al., 2009; Mester et al., in press).
In their study (IGDA, 2004) the International Game Developers Association discusses the availability of games to every person with a disability. The study presents speech synthesisers (TTS), screen readers and speech recognition (ASR) as assistive technologies which can contribute to a greater availability of games for the visually impaired. Unfortunately, the application of these new technologies which would allow the adaptation of the user interface to the ergonomics of the visually impaired is not the primary concern for the game industry. On the other hand, a connection between the visually impaired children and these technologies is of a crucial importance to their inclusion into the society (Perepatić, 2010).
User interfaces differ from game to game, and consequently, the ability to be adapted to the visually impaired, as well as the process of adaptation itself, also differs from game to game. Audio signals are the key factor of these games when adapted to the visually impaired. These audio signals have to differ from each other so that the player can easily identify them and respond at the right moment and in a right manner. At http://www.AudioGames.net,

one of the best known web sites with audio games, there are more than 300 audio games, and their classification and examples are given in (Mester et al., in press).

In the process of creation of audio games particular attention is paid to presenting information in audio form, because sound presentation must carry all relevant information that allows the player to react timely and in the right manner. The GUI of a video game carries most of the information, which gives particular broadness and freedom while developing such games as opposed to audio games. Portraying all relevant information in audio form presents an interesting challenge because the presentation of audio information to the user is limited. In sound-based games the player gets a mental picture of all present objects and persons by listening to the sounds which characterise them. Stereo positioning is used to spatially distinguish the sounds of objects. It allows the sound to traverse from left to right and vice versa. These sounds are critical for the player and his/her understanding of the game. Yet stereo positioning only gives the player one dimension, which is a constraint compared to the two dimensions of a screen.

For example, Delić & Vujnović Sedlar (2010) have created the first audio game for the visually impaired with ASR and TTS in Serbian. It is a simple memory game with sixteen fields hiding eight pairs of objects. Having in mind the characteristics of binaural hearing the authors have decided to present the horizontal position of the field by simple stereo presentation (different interaural levels between ears), and to indicate the vertical position of the field by using different audio frequencies (pitch of synthesised speech – TTS) similarly to (Gärdenfors, 2003). The user has a sensation of sound coming from an exact position on a four-by-four grid facilitating memorisation of object locations. The user can select the square either using verbal commands (by pronouncing the coordinates of the square – ASR) or simply using the keyboard. The memory game has been developed as Microsoft application in C#, using Microsoft Visual Studio 2008, with sound supported by Microsoft DirectX SDK.

Another example of a computer game suitable to be adjusted to the visually impaired using audio and speech technologies is a set of very simple geometric puzzles named Lugram (Lučić & Vujnović Sedlar, 2009). Geometry as a branch of mathematics is one of the most difficult areas from the point of view of adaptation for the visually impaired, but on the other hand, it is very useful for orientation in space and executing everyday tasks. Following the example of the ancient Chinese Tangram, Lugram has been designed as a puzzle game aimed at composing given geometric figures. Elements to be used for assembling are square tiles containing geometric figures such as triangles, rectangles or squares, as shown in Fig. 10. One direction of the development of the game led to its successful adaptation for visually impaired users (Lučić et al., 2009), and opened the perspective for a special challenge of creating a new version of the game for the blind. Lugram has been developed using C++ and Macromedia's Director.

The audio interface of the game consists of speech, music and various sorts of audio effects. Speech is mostly used to introduce the user to the guidelines and rules of the game, and for this purpose TTS is most commonly used. Synthesised speech can be used in the game itself more or less, depending on the type of the game. Generally, if greater authenticity of the situation is to be achieved, or in order to ensure that the right reaction will be made, in most cases synthesised speech can be replaced by recorded – natural speech. Audio effects are commonly used to illustrate situations or various objects in the game (Ratanasit & Moore, 2005). For the visually impaired it is of particular importance to have certain audio effects which would tell them whether their reaction was suitable or not. Music can also be a good element for depicting states and situations a player can find himself/herself in, or it can be used just as a background.

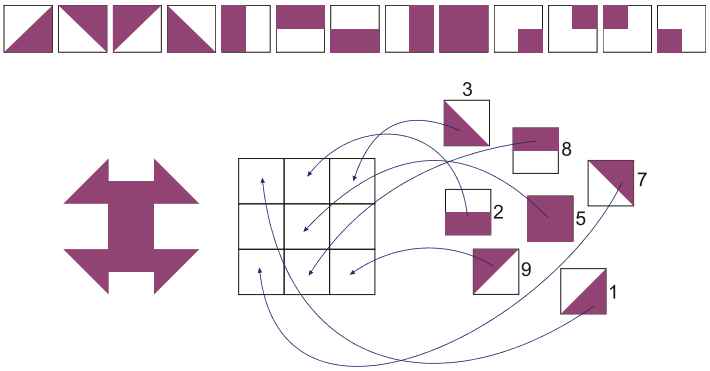


Fig. 10. The squares containing geometric figures and an example of a task in the computer game Lugram intended for the visually impaired

Due to the lack of sight, the blind rely on other senses heavily, especially on the senses of touch and hearing, making them more advanced (Doucet, 2005), and allowing them to easily learn to use keyboards very skillfully. The alternative to using keyboards is ASR. Because of intra- and interpersonal differences in the voices of speakers, different setups and qualities of microphones and the communication channels, as well as different levels of ambient noise, ASR is a very demanding task for the computer games and is not well developed for all languages. Studies usually mention using ASR for issuing certain voice commands, but unrestricted human-to-computer speech communication is not so common yet.
Audio interfaces enable the visually impaired to play games more equally to other players. As speech is an extremely important element of such an audio interface, speech technologies are essential for playability of games with audio interfaces. Development of speech technologies is thus a contribution to inclusion of persons with disabilities into the society.

5. Conclusion

The applications presented in this chapter clearly show the importance of development of speech technologies. Having in mind the extreme language dependence of these technologies, and the fact that, unlike most other technologies, they cannot simply be „imported from abroad", it is very important that scientific teams from the region should be actively engaged in their research and development. Only thus we can expect that the 20 million inhabitants of this part of Europe will be able to communicate with machines by speech in their native languages in a near future.

5.1 Directions of further research and development

One of the directions of furher research and development of speech technologies is multilingual and multimodal human-computer interaction involving not only ASR and TTS but speaker and emotion recognition as well. Besides further research aimed at increasing the quality of ASR and TTS components, research related to implementation of speech technologies on embedded platforms is also under way, aimed at their application in small portable devices. ASR and TTS have an extremely wide area of application, and some projects are initiated to apply developed speech technologies in South Slavic languages in smart homes, cars, industry, robots and toys. This would enable a number of other applications

such as dictation, automated transcription of radio and TV programmes, meetings and sessions, telephone conversations etc.

6. References

Delgado, R. & Araki, M. (2005). Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment. Wiley, ISBN: 978-0-470-02155-2

Delić, V.; Sečujski, M. & Pekar, D. (2005). On anReader, its features and first applications (in Serbian), *Computers and the Blind*, Zagreb, Croatia

Delić, V.; Sečujski, M. & Tekić, Ž. (2006). A Contribution to Human-Machine Communication in Serbian, Croatian and Macedonian Language, *Proceedings of 17th DAAAM Symposium (Intelligent Manufacturing&Automation: Focus on Mechatronics&Robotics)*, DAAAM Publ., Vienna, TU Wien, pp. 101-102, ISSN 1726-9679, ISBN 3-901509-57-7

Delić, V. (2007). A Review of R&D of Speech Technologies in Serbian and their Applications in Western Balkan Countries, *Keynote lecture at 12th SPECOM (Speech and Computer)*, pp. 64-83, ISBN 6-7452-0110-x, Moscow, Russia, October 2007

Delić, V. & Vujnović Sedlar, N. (2010). Stereo Presentation and Binaural Localization in a Memory Game for the Visually Impaired. *Lecture Notes in Artificial Intelligence, Springer,* A. Esposito et al. (Eds.): COST 2102 Int. Training School 2009, LNAI 5967, pp. 354-363. Springer, Heidelberg, ISSN 0302-9743

Delić, V.; Sečujski, M.; Jakovljević, N.; Janev, M.; Obradović, R. & Pekar, D. (2010). Speech Technologies for Serbian and Kindred South Slavic Languages, Chapter in the book *Speech Recognition,* SCIYO, ISBN 978-953-7619-X-X (accepted for publication)

Doucet, M.-E.; Guillemot, J.-P.; Lassonde, M.; Gagné, J.-P.; Leclerc, C. & Lepore, F. (2005). Blind subjects process auditory spectral cues more efficiently than sighted individuals. *Exp. Brain Res.* 160: 194–202

Gärdenfors, D. (2003). Designing Sound-Based Games. *In Digital Creativity,* 14(2), 111-114

IGDA - International Game Developers Association. (2004). Accessibility in Games: Motivations and Approaches. Retrieved on February 15, 2005, from http://www.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf

Janev, M.; Pekar, D.; Jakovljević, N. & Delić, V. (2008). Eigenvalues driven gaussian selection in continuous speech recognition using HMMs with full covariance matrices. *Applied Intelligence, Springer Netherlands,* DOI: 10.1007/s10489-008-0152-9, (Print, accepted) December 2008, ISSN 0924-669X (Print) 1573-7497 (Online), Available at: http://www.springerlink.com/content/964vx4055k424114/

Lučić, B. & Vujnović Sedlar, N. (2009). Geometric Puzzle Lugram - Development and Application (in Serbian). In *Proceedings of TELFOR: Vol. 17.* Belgrade

Lučić, B.; Vujnović Sedlar, N. & Delić, V. (2009). Computer game Lugram - version for visually empaired children (in Serbian). In *Proceedings of TELFOR: Vol. 17.* Belgrade

Mester, Gy.; Stanić-Molcer, P. & Delić, V. (in press). Educational Games, A chapter in the book *Business, Technological and Social Dimensions of Computer Games: multidisciplinary developments*, Publisher: IGI Global, PA, USA (accepted for publication)

Mišković, D.; Đurić, N.; Pekar, D. & Jakovljević, N. (2007). Alfanum word spotter as a form of ASR application. *Proceedings of 51th ETRAN*, Herceg Novi – Igalo, June 4 – 8, 2007

Mišković, D.; Zindović M. & Pekar D. (2007). Postprocessing methods for validation of Alfanum ASR recognition system. *Proceedings of TELFOR,* Beograd

Mišković, D.; Vujnović, N.; Sečujski, M. & Delić, V. (2005). Audio Library for the Visually Impaired as an Application of TTS Tehnology (in Serbian). *Proceedings of 49th ETRAN,* Vol II, pp. 400-402, ISBN 86-80509-54-X, Budva, Montenegro, Publisher: Society for ERAN

Mišković, D.; Vujnović, N.; Sečujski, M. & Delić, V. (2006). Audio Library for the visually impaired – ABSS 2.0. *Proceedings of DOGS (Digital Signal and Image Processing)*, pp. 67-70, Vršac, Serbia, Publisher: Faculty of Technical Sciences, Novi Sad.

Nöth, E.; Horndasch, A.; Gallwitz, F. & Haas, J. (2004). Experiences with Commercial Telephone-based Dialogue Systems. *it – Information Technology*, Vol. 46, No. 6, 306-314, ISSN: 1611-2776

Pekar, D.; Delić, V.; Molerov, S.; Kočiš, G. & Vuković, R. (2007). System for automatic recognition of audio clips in radio and TV programmes, Patent in Serbia P-2007/0505

Perepatić, J. (2010). Possible benefits of computer games to the visually impaired children - a survey of parents' opinions. Non-Government Organisation "Iskrica", Novi Sad, Serbia (unpublished).

Ratanasit, D. & Moore, M. M. (2005). Representing Graphical User Interfaces with Sound: A Review of Approaches. *Journal of Visual Impairment and Blindness,* 99(2), 69-84.

Ronto, R.; Pekar, D.; Đurić, N. (2005). Developing a Telephone Voice Portal with ASR and TTS Capability (in Serbian). *Proceedings of 49th ETRAN,* Tom II, pp. 392-395, ISBN 86-80509-54-X, Budva, Montenegro, June 2005, Publisher: Society for ERAN. Portal is available at: http://www.alfanum.ftn.uns.ac.rs/kontakt/

Sečujski, M.; Obradović, R.; Pekar, D.; Jovanov, Lj. & Delić, V. (2002). AlfaNum System for Speech Synthesis in Serbian Language. *Proceedings of TSD (Text, Speech and Dialogue)*, ISBN 3-540-44129-8, Brno, Czech Republic, September 2002. *Lecture Notes in Artificial Intelligence*, *Springer-Verlag Berlin Heidelberg,* LNAI 2448, pp. 237-244, ISSN 0302-9743

Sečujski, M.; Delić, V.; Pekar, D.; Obradović, R. & Knežević, D. (2007). An Overview of the AlfaNum Text-to-Speech Synthesis System, *Proceedings of 12th SPECOM (Speech and Computer)*, pp. Ad.Vol. 3-7, ISBN 6-7452-0110-x, Moscow, Russia, October 2007, Demo is available at: http://www.alfanum.co.rs/anreader.html

Žganec Gros, J.; Delić, V.; Pekar, D.; Sečujski, M. & Mihelič, A. (2006). The iTEMA E-mail Reader, *Proceedings of IS-LTC*, pp. 230-233, Ljubljana, Slovenia, ISSN 1581-9973, ISBN-13 978-961-6303-83-X.

Žganec Gros, J.; Delić, V. & Pekar, D. (2008). Listen to your e-mail through the telephone, *eStrategies|Projects EUROPE*, Vol. 2, No. 3, pp. 42-43, British Publishers, ISSN 1752-5152.

8

Croatian Speech Recognition

Ivo Ipšić and Sanda Martinčić–Ipšić
University of Rijeka
Croatia

1. Introduction

In the chapter we describe procedures for Croatian speech recognition which are used in a limited domain spoken dialog system for Croatian speech. The dialog system would provide information about weather in different regions of Croatia for different time periods (Žibert et al., 2003). The spoken dialog system includes modules for automatic speech recognition (ASR), spoken language understanding and text-to-speech synthesis. In this work ASR module based on data-driven statistical and rule-based knowledge approach is discussed. Data driven statistical approach is based on large quantities of spoken data collected in the speech corpus. Rule based approach is based on Croatian linguistic and phonetic knowledge. Both approaches must be combined in a spoken dialog system because there is not enough speech data to statistically model the human speech and there is not enough knowledge about the processes in human mind during speaking and understanding (Dusan & Rabiner, 2005). Speech recognition today, as in the past decades, is mainly based on data driven statistical approaches (Huang et al. 2000; Rabiner, 1989). Statistical pattern recognition and segmentation algorithms and methods for stochastic modelling of large speech quantities are used. The data driven statistical approach uses hidden Markov models (HMM) as the state of the art formalism for speech recognition. Many large vocabulary automatic speech recognition systems (LVASR) use mel-cepstral speech analysis, hidden Markov modelling of acoustic sub word units, n-gram language models (LM) and n-best search of word hypothesis (Furui, 2005; O'Shaugnessy, 2003; Huang et al., 2000; Jelinek, 1999). Speech recognition research in languages like English, German and Japanese (Furui et al., 2006) has focus in recognition of spontaneous and broadcasted speech. For highly flective Slavic and agglutinative (Kurimo et al., 2006) languages the research focus is still more narrowed mainly due to the lack of speech resources like corpuses. Large or limited vocabulary speech recognition for Slovene (Žibert et al., 2003), Czech (Lihan et al., 2005; Psutka et al., 2003), Slovak (Lihan et al., 2005), Lithuanian (Skripkauskas & Telksnys, 2006; Vaičiūnas & Raškinis, 2005) and Estonian (Alumäe & Võhandu, 2004) with applications for dialog systems (Žibert et al., 2003), dictation (Psutka et al., 2003) or automatic transcriptions (Skripkauskas & Telksnys, 2006) have been reported lately.

Croatian is a highly flective Slavic language and words can have 7 different cases for singular and 7 for plural, genders and numbers. The Croatian word order is mostly free, especially in spontaneous speech. The unstressed word system is complex because the possible transition of the accent from a stressed word to the unstressed one is conditioned by the position of the word in a sentence, which is mostly free. Standard Croatian

pronunciation rules sometimes allow more different word accents. Mostly free word order, a complex system of unstressed words and nondeterministic pronunciation rules make the development of pronunciation dictionary and prosodic rules difficult. On the other hand Croatian orthographic rules based on phonological-morphological principle are quite simple which simplifies the definition of orthographic to phonetic rules and process of phonetic transcription.

The number of Croatian native speakers is less then 6 millions. Still some interest in the research and development of speech applications for Croatian can be noticed. The speech translation system DIPLOMAT between Serbian and Croatian on one side and English on the other is reported in (Frederking, et al., 1997; Scheytt, et al., 1998; Black, et al., 2002). The TONGUES project continued with this research in direction towards large Croatian vocabulary recognition system.

Croatian orthographic-to-phonetic rules are proposed for phonetic dictionary building. The developed Croatian multi-speaker speech corpus was successfully used for the development of speech applications. Proposed Croatian phonetic rules captured adequate Croatian phonetic, linguistic and articulatory knowledge for state tying in acoustical models of the speech recognition system.

The Croatian speech recognition system is based on continuous hidden Markov models of context independent (monophones) and context dependent (triphones) acoustic models. The training of speech recognition system was performed using the HTK toolkit (Young et al., 2002; HTK, 2002).

Since the main resource in a spoken dialog system design is the collection of speech material, the Croatian speech corpus is presented in Section 2. Orthographic-to-phonetic rules used in the phonetic dictionary preparation are shown as well. Further the acoustic modeling procedures of the speech recognition system including phonetically driven state tying procedures are given in Section 3. Conducted speech recognition experiments and speech recognition results are presented in section 4. We conclude with discussion on advantages of the proposed acoustical modelling approach for Croatian speech recognition and description of current activities and future research plans.

2. The Croatian speech corpus

The Croatian speech corpus includes news, weather forecasts and reports spoken within broadcasted shows of the national radio and television news broadcasted at the national TV (Martinčić-Ipšić and Ipšić, 2004). The collected speech material is divided into several groups: weather forecasts read by professional speakers within national radio news, weather reports spontaneously spoken by professional meteorologists over the telephone, other meteorological information spoken by different reporters and daily news read by professional speakers.

The speech corpus is a multi-speaker speech database which contains 16,5 hours of transcribed speech spoken in the studio acoustical environment and 6 hours of telephone speech. The spoken utterance has its word level transcription.

The first part of the speech corpus consists of transcribed weather forecasts and news recorded from the national radio programmes. This is a multi-speaker database, which contains speech utterances of 11 male and 14 female professional speakers. The radio part consists of 9431 utterances and lasts 13 hours. The transcribed sentences contain 183000 words, where 10227 words are different. Relatively small number of 1462 different words in

the weather forecast domain shows that this part of the speech database is strictly domain oriented.
The second part contains weather reports given by 7 female and 5 male professional meteorologists over the telephone. The 170 transcribed weather reports are lasting 6 hours and contain 1788 different words in 3276 utterances. Most of the speech captured in the telephone part can be categorized as semi-spontaneous. This data is very rich in background noises such as door slamming, car noise, telephone ringing and background speaking and contains noise produced by channel distortions and reverberations. All this special events and speech disfluencies and hesitations are annotated in transcriptions by < >.
The third part of the speech database consists of TV News broadcasted at the national TV – HTV. The news data is not domain oriented. Diversity of subjects and topics is noticeable in the number of all words compared to the number of different words. Further the number of speakers is also significantly bigger then in the weather part of the database. The news data is also very rich in different background noises, including music, it also contains commercials, reports in foreign languages and so on. All of this features where captured and annotated during the transcription. The transcribed part of TV News consists of 18632 words where 9326 are different. The transcribed part of TV News is 3 hours and 28 minutes. long. Statistics of TV News is also shown in the bottom part of Table 1.

	Number		Speakers		Words		Dur.
	Reports	Utts.	Male	Fem.	All	Diff.	[min]
Radio weather forecasts	1057	5456	11	14	77322	1462	482
Radio news	237	3975	1	2	105678	9923	294
Overall RADIO	1294	9431	11	14	183000	10227	775
Teleph. weather reports	170	3276	5	7	52430	1788	360
BCN	6	280	217		18632	9326	208
Overall	1470	12987	253		254062	15998	1343

Table 1. Croatian speech corpus statistics.

2.1 Data acquisition and transcription

The broadcasted radio news with weather forecasts and telephone weather reports were recorded four times a day using a PC with an additional Haupage TV/Radio card. The speech signals are sampled with 16 kHz and stored in a 16-bit PCM encoded waveform format. At the same time texts of weather forecasts for each day were collected from the web site of the Croatian Meteorological Institute. The texts were used for speech transcription and for training of a bigram language model for the weather forecast speech recognition system. For the telephone weather reports and daily news no adequate text existed so the whole transcription process was manual. The transcribing process involved listening to speech until a natural break was found. The utterances or parts of speech signals were cut out and a word level transcription file was generated. The speech file and the transcription file have the same name with different extensions.
During the transcription some basic rules were followed: all numbers and dates were textually written, all acronyms and foreign names were written as pronounced and not as

spelled and all other words were written according to the Croatian writing rules (Anić and Silić, 2001). Word transcriptions of TV news have been done in two stages. In the first stage we collected texts from TV NEWS at the internet site of the national TV (HTV). The texts were not the exact transcriptions and we had to correct them, but they were a good start.
All final transcriptions of Croatian BCN (Broadcast News) were made with the Transcriber tool (Barras, et al., 2000). Transcriber is a tool for assisting in the creation of speech corpora enabling manual segmentation and transcription as well as annotation of speech turns, topic and acoustic condition. The data format follows the XML standard with Unicode support for multilingual transcriptions (Graff, 2000).

2.2 Phonetic dictionary

For the word segmentation and recognition task we have developed a phonetic dictionary, where we proposed a set of phonetic symbols to transcribe the words from the Croatian speech database. The selected symbols are derived according to the Speech Assessment Methods Phonetic Alphabet (SAMPA) (SAMPA, 1997). The standard phoneme set includes 30 phonemes, where the set of vowels is extended with the vibrant vowel /r/.
Croatian orthographic rules are based on the phonological-morphological principle which enables automatisation of phonetic transcription. Standard definition of orthographic to phonetic rules, one grapheme to one phonetic symbol was extended with additional rules for example:

- words with group ds were phonetically transcribed as [c] and
- words with suffixes naest were phonetically transcribed as [n a j s t].

The phonetic dictionary comprises all words in transcription texts. All word forms (different cases, genders and numbers of the same basic word form) are considered as a new word in the dictionary. The current phonetic dictionary contains 15998 different words. The fact that Croatian language is highly flective reflects to the size of the phonetic dictionary. The dictionary can contain few entries for the same basic word format. For example the word bura, which denotes the northern wind type, is represented by 4 different word forms: bura, bure, burom, buru. Since all foreign names were written as pronounced there was no need for writing the orthographic to phonetic rules for languages like English, German, Italian, Chinese, Arab, etc.
The accent position is embedded in the dictionary with differentiation between accented and non–accented vowels. For the words that can be pronounced in more correct ways the position of the really accented vowel was marked.

2.3 Segmentation

Since the transcription of the speech files is on the word level for the training procedures the utterances have to be segmented on the phone level. The initial segmentation is performed using automatic alignment of speech signals and word transcriptions, which is based on hidden Markov monophone models. The automatic segmentation is performed using the monophone speech recognizer described in section 3.
Typical segmentation errors detected during manual inspections of automatically determined speech segments can be roughly classified as transcription errors and real segmentation errors. Similar automatic segmentation error taxonomy for English is presented in (Kominek, et al., 2003).
Transcription errors are errors in the speech transcription stage of speech corpora development. Some words or special acoustic events were incorrect or inaccurate typed or

were not typed at all. For example if breathing noise (inspiration) was not marked in the textual transcription in a utterance, the whole inspiration was segmented as a really long phoneme.
Real segmentation errors occurred when transcriptions were correct but the segment interval was not determined correctly. Typical segmentation errors occurred:

- at infrequent phones like /lj/ or /dž/,
- at two following vowels which are seldom in Croatian words like /ea/ and
- at too tightly segmented phones combinations where one of the phones was not pronounced like /ije/.

Automatically segmented speech utterances were manually inspected and segmentation errors were corrected in the speech database.

3. Acoustic and language modelling

The goal of speech recognition system is to recognize the spoken words represented by a stream of input feature vectors calculated from the acoustic signal. The major problems in continuous speech recognition arise due to the nature of spoken language: there are no clear boundaries between words, the phonetic beginning and ending are influenced by neighbouring words, there is a great variability in different speakers speech: male or female, fast or slow speaking rate, loud or whispered speech, read or spontaneous, emotional or formal and the speech signal can be affected with noise. To avoid these difficulties the data driven statistical approach based on large quantities of spoken data is used (Furui et al., 2006). Statistical pattern recognition and segmentation algorithms and methods for stochastic modelling of time varying speech signals are used (Rabiner et al., 1989; Huang et al., 2000; Duda et al., 2001). Additionally statistical language models are used in order to improve the recognition accuracy (Jelinek et al., 1999).
The data driven statistical approach uses hidden Markov models (HMM) as the state of the art formalism for speech recognition. Hidden Markov models are stochastic finite-state automata consisting of finite set of states and state's transitions. The state sequence is hidden, but in each state according to the output probability function an output observation can be produced.
The HMM Φ is defined by a triplet $\Phi=(A,B,\Pi)$ where A is state transition probability matrix, B is speech signal feature output probability matrix and Π is the initial state probability matrix. The output probability density function is represented by a mixture of Gaussian probability density function $b_j(x)=N(x,\mu_{jk},\Sigma_{jk})$ (Huang et al., 2000)

$$b_j(x) = \sum_{k=1}^{M} c_{jk} N(x, \mu_{jk}, \Sigma_{jk}) = \sum_{k=1}^{M} c_{jk} b_{jk}(x) \quad \text{for } j = 1..N \text{ and } t = 1..T, \tag{1}$$

where

x	is the speech signal feature vector,
$b_j(x)$	is a Gaussian probability density function associated with state s_j,
μ_{jk}	is mean vector of the k^{th} mixture in state s_j,
Σ_{jk}	is covariance matrix of the k^{th} mixture in state s_j
M	is the number of mixture components and
c_{jk}	is the weight for the k^{th} mixture in state s_j satisfying the condition:

$$\sum_{k=1}^{M} c_{jk} = 1, \text{ and } c_{jk} \geq 0, \quad 1 \leq j \leq N, \ 1 \leq k \leq M. \tag{2}$$

For the estimation of continuous HMM parameters iterative Baum-Welch procedure is used. The in Baum-Welch also known as the Forward-Backward algorithm iteratively refines the HMM parameters by maximizing the likelihood of a speech signal feature sequence X given a HMM Φ, $P(X|\Phi)$. The algorithm is based on the optimisation technique used in the EM algorithm for the estimation of Gaussian mixture densities parameters. The Baum-Welch algorithm uses iteratively forward and backward probabilities which define the probability of the partial observation sequence X_t at time t in state i, given the HMM Φ (Duda et al., 2001; Huang et al., 2000).

For the search of an optimal path in the HMM network of acoustic models the Viterbi algorithm is used (Rabiner, 1989). Viterbi algorithm is a dynamic programming algorithm that decodes the state sequence according to the observed output sequence.

For speech modelling and recognition the speech signal feature vectors consist of 12 mel-cepstrum coefficients (MFCC), frame energy and their derivatives and acceleration coefficients. The feature coefficients were computed every 10 ms for a speech signal frame length of 20 ms.

Figure 1 presents main steps performed in the Croatian speech recognition system development, where acoustic and language models are trained. The speech signal is parameterized with MFCC feature vectors and their dynamic components, where the spectral resolution of the human ear is modelled. Speech transcriptions and speech signal feature vectors are used to train parameters of the monophone HMMs. The automatic segmentation is performed using monophone HMMs. The results of automatic segmentation are time intervals for each spoken phone. The automatically segmented phones are used for training (estimating) the parameters of monophone HMMs by repeating the Baum-Welch re-estimation procedure. The training procedure is repeated for each increase of the Gaussian mixture component. The triphones are constructed from monophones in a way that each triphone has in the left and in the right context the preceding and the succeeding phone. The triphone HMMs are constructed from monophone HMMs and the parameters are estimated with the Baum-Welch procedure.

The triphone states with estimated parameters value are tied according to the proposed Croatian phonetic rules. The state tying procedure insures enough acoustic material to train all context dependent HMMs and enables acoustic modelling of unseen acoustic units, that are not present in the training data. The parameters of tied triphone HMMs are estimated by repeating the Baum-Welch re-estimation procedure and by increasing the number of Gaussian mixtures. The prepared textual transcriptions of speech utterances and phonetic dictionary are used to build a bigram language model. The triphone HMMs and bigram language model are used for Croatian speech recognition.

The acoustic model should represent all possible variations in speech. Variations in speech can be caused by speaker characteristics, coarticulation, surrounding acoustical conditions, channel etc. Therefore selection of an appropriate acoustic unit, which can capture all speech variations, is crucial for acoustic modelling. Enough acoustic material should be available for HMMs modelling of chosen acoustic unit. At the same time the chosen acoustic unit should enable construction of more complex units, like words (Odell, 1995). In continuous speech recognition systems the set of acoustic units is modelled by a set of HMMs. Since the

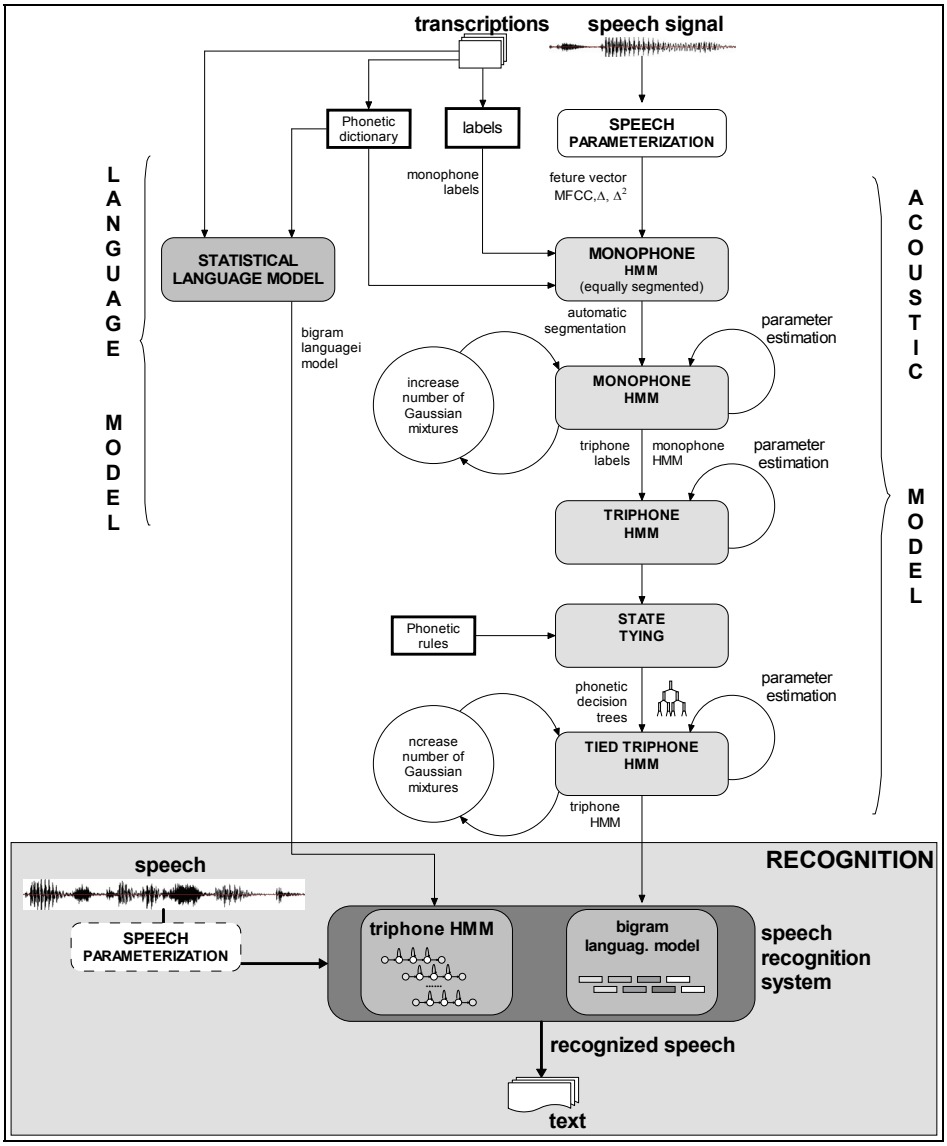


Fig. 1. Development of the Croatian speech recognition system.

number of units is limited (by the available speech data) usually the subword acoustic units are modelled. The subword units are: monophones, biphones, triphones, quinphones (Gauvain & Lamel, 2003; Lee et al., 1990) or sub phonemic units like senones (Hwang et al., 1993). Some speech recognition systems are modelling syllables (Shafran & Ostendorf, 2003) or polyphones (Schukat-Talamazzini, 1995). All these units are enabling construction of the more complex units and recognition of the units not included in the training procedure (unseen units).

3.1 Context independent acoustic model

The training of speech recognition acoustic models started with defining the Croatian phoneme set according to SAMPA (SAMPA, 1997). For each of 30 Croatian phonemes a context independent monophone hidden Markov model was defined. Initially the monophone models with continuous Gaussian output probability functions described with diagonal covariance matrices were trained. Each monophone model consists of 5 states, where the first and last states have no output functions. The initial training of the Baum-Welch algorithm on HMM monophone models resulted in a monophone recognition system, which was used for the automatic segmentation of the speech signals. The automatic segmentation of the speech signal to the phone level is performed using the forced alignment (Young et al., 2002) of the spoken utterance and the corresponding word level transcriptions. The results of automatic segmentation are exact time intervals for each phone. Further, the monophone models were trained by 10 passes of the Baum-Welch algorithm and the resulted monophone models were used for the initialization of context dependent triphone hidden Markov models. The number of mixtures of output Gaussian probability density functions per state was increased up to 20.

3.2 Context dependent acoustic model

The triphone context-dependent acoustic units were chosen due to the quantity of available speech and possibility for modelling both, left and right, coarticulation context of each phoneme. We trained context-dependent cross-words triphone models with continuous density output functions (up to 20 mixture Gaussian density functions), described with diagonal covariance matrices. The triphone HMMs consist of 5 states, where the first and last states have no output functions.

Table 2 shows the number of cross-word seen triphones in the training data used for radio speech recognition training. Evidently there was not enough acoustical material for modelling all possible triphone models. The severe under training of the model can be a real problem in the speech recognition system performance (Hwang et al., 1993). The lack of speech data is overcome by a phonetically driven state tying procedure.

	No.		No. triphones		%
	monophones	possible	all	seen	seen
radio weather	29+4	35937	31585	4042	12.80%
radio news	30+4	39304	36684	7931	21,62%
telephone	29+4	35937	31585	4618	14.62%

Table 2. The number of monophones and triphones and seen triphones percentage per parts of the speech corpus.

3.3 Croatian phonetic rules and decision trees

The state tying procedure proposed in (Young et al., 1994) allows classification of unseen triphones in the test data into phonetic classes and tying of the parameters for each phonetic class. In our system 108 phonetic rules (216 Croatian phonetic questions about left and right context (Martinčić-Ipšić & Ipšić, 2006a)) are used to build phonetic decision trees for HMM state clustering of acoustic models. The phonetic rules are describing the classes of the phonemes according to their linguistic, articulatory and acoustic characteristics. A phonetic decision tree is a binary tree, where in each node the phoneme's left or right phonetic

context is investigated. The phonemes are classified into phonetic classes depending on the phonetic rules which examine the phoneme's left and right context. Some Croatian phonetic rules used for the training of phonetic classes are shown in Table 3.

Vowel	a, e, i. o, u
High Vowel	i, u
Medium Vowel	o, e
Back	k, g, h, o, u
Affricate	c, C, cc, dz, DZ
Velar	k, g, h
Glide	j, v
Apical	t, d, z, s, n, r, c, l
Strident	v, f, s, S, z, Z, c, C, DZ
Constant Consonant	v, l, L, j, s, S, z, Z, f, h
Unvoiced Fricative	f, s, S, h
Compact Consonant	N, L, j, S, Z, C, cc, dz, DZ, k, g, h

Table 3. Examples of Croatian phonetic classes.

Figure 2 presents an example of phonetic decision tree for Croatian phoneme /h/. It classifies triphones with the phoneme /h/ in the middle in eight possible classes. At each node the binary question (from the set of 108 phonetic rules) about left and right context is asked and YES/NO answers are possible. The triphones in the same class are sharing the same parameters (state transition probabilities and output probability density functions of HMMs).

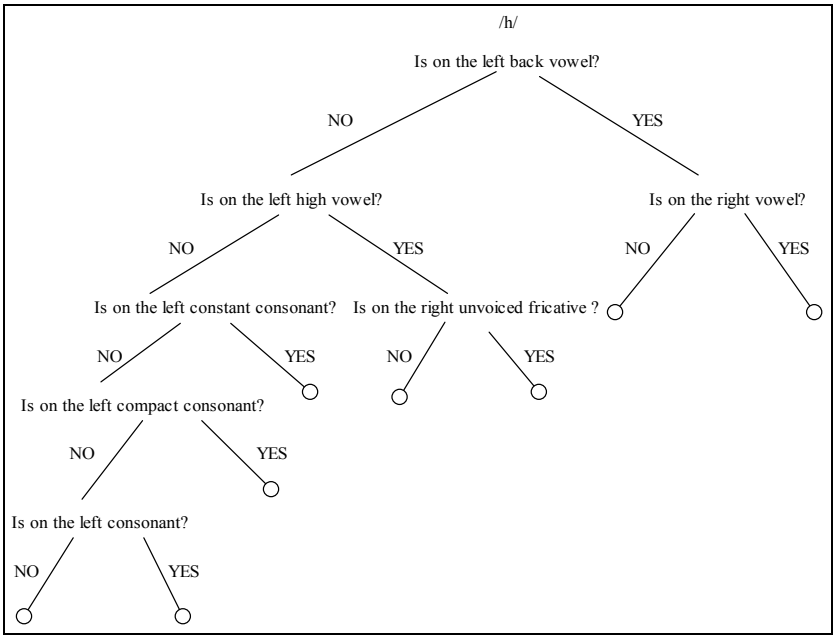


Fig. 2. The decision tree of phonetic questions for the left and right context of phoneme /h/.

For the construction of the phonetic decision tree from phonetic rules and from parameters of triphone HMM states a state tying procedure proposed in (Young et al., 1994) is used. Tying enables clustering of the states that are acoustically similar, which allows all the data associated with one state to be used for more robust estimation of the model parameters. This enables more accurate estimating mixtures of Gaussian output probabilities and consequently better handling of the unseen triphones.
For each phoneme a decision tree is built using a top-down sequential optimization procedure (Odell, 1995). Initially all states are placed in the root node. So, all states are initially tied together and log likelihood is calculated for this node. The tying procedure iteratively applies phonetic rules to the states of the triphone models and partitions the states into subsets according to the maximum increase in log likelihood. When the threshold is exceeded the tied states are no further partitioned.
State tying enables clustering of the states that are acoustically similar, which allows all the data associated with one state to be used for more robust estimation of the model parameters (mean and variance). This enables more accurate estimation of Gaussian mixtures output probabilities and consequently better handling of the unseen triphones.
For the speech recognition task the state clustering procedure uses a separate decision tree for initial, middle and final states of each triphone HMM which is built using a top-down sequential sub-optimal procedure (Odell, 1995). Initially all relevant states are placed in the root node. So, all states are initially tied together and log likelihood is calculated for this node. The tying procedure iteratively applies phonetic rules to the states of the triphone models and partitions the states into subsets according to the maximum increase in log likelihood. When the threshold is exceeded the tied states are no further partitioned.
For a set S of HMM states and a set F of training vectors x the log likelihood L(S) is calculated according to (Young et al., 1994) by

$$L(S) = \sum_{f=1}^{F} \sum_{s=1}^{S} \log(P(x_f, \mu(S), \Sigma(S))) \xi_s(x_f), \tag{3}$$

where $P(x_f,\mu(S),\Sigma(S))$ is the probability of observed vector x_f in state s under the assumption that all tied states in the set S share a common mean vector $\mu(S)$ and variance $\Sigma(S)$. $\xi_s(x_f)$ is the posterior probability of the observed feature vector x_f in state s and is computed in the last pass of the Baum-Welch re-estimation procedure (Young et al., 2002).
The node with states from S is partitioned into two subset S_y and S_n using phonetic question Q which maximizes the ΔL:

$$\Delta L = L(S_y) + L(S_n) - L(S), \tag{4}$$

where S_y is set of states which are satisfying the investigated phonetic question Q and in the S_n set are the rest of the states. Further the node is split according to the phonetic question which gives the maximum increase in log likelihood. The procedure is then repeated until it exceeds the threshold. The terminal nodes share the same distribution so the parameters of the final nodes can be estimated accurately, since the tying procedure provides enough training data for each final state.
The state tying procedure is presented in figure 3. From the top first is shown a monophone HMM for phoneme /h/. At the second level are HMMs for triphones o-h+r, e-h+a and a-h+m. Then the triphone states where tied and states sharing the same parameters are clustered using the phonetic decision trees. And at the bottom are the same tied states with

increased number of mixtures of Gaussians probability functions evaluated by the Baum-Welch parameter reestimation procedure.

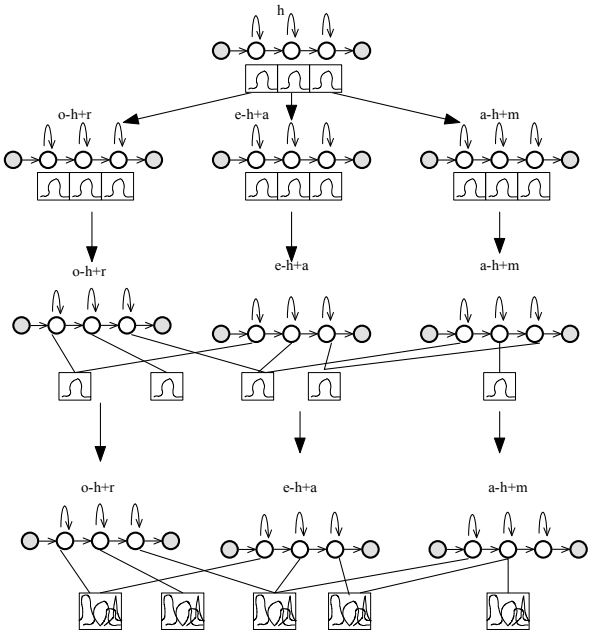


Fig. 3. The state tying procedure for the triphones with /h/ in the middle.

Table 4 contains the most frequently used Croatian phonetic questions in the phonetic decision trees in the speech recognition systems. Phonetic questions in the table are abbreviated. For instance the R-Front is the abbreviated phonetic question: Is the phoneme in the right context from the articulatory class front? Phonetic questions are ranked according to the appearance frequency in the decision trees. For the speech recognition part the frequency is calculated over 3 different sets of phonetic trees with different number of tied states (clusters).

Radio speech		Telephone speech	
Phonetic question	No.	Phonetic question	No.
R_Front	811	R_Front	522
L_Front	797	L_Front	498
L_Vowel-Open	635	L_Central	348
L_Central	594	R_Vowel-Open	336
R_Vowel-Open	561	R_Central	312
L_Consonant-Voiceless	432	L_Vowel-Open	312
R_Vowel	384	L_Consonant-Voiceless	222
R_Consonant-Voiceless	357	R_Vowel	221
D_Central	355	D_Consonant-Voiceless	216
L_Nasal	338	L_Consonant-Closed	201

Table 4. The most frequently used Croatian phonetic questions in radio and telephone speech recognition.

As expected and reported for other languages (Gauvain & Lamel, 2003) the most common Croatian phonetic rules (front, central, vowel) are the most frequently used for phonetic clustering in the speech recognition system. Since the results are presented for left and right coarticulation context and for the stable part of the phoneme, the phonetic rules are in left-question, right-question pairs. Phonetic questions investigating the presence of the single phoneme in the coarticulated context are the less frequent one, and used only in phonetic trees with higher number of tied states.

3.4 Language modelling

Language model is an important part of the speech recognition system. The language model estimates the probabilities of word sequences which are derived from manual transcriptions of the speech database and from normalized text corpora. In this work statistical language model was used (Jelinek, 1999). N-gram statistical language models are modelling the probability P(W) for the sequence of words $W=w_1,w_2,..,w_n$

$$P(W) = \prod_{i=1}^{n} P(w_i \mid w_1, w_2, .., w_{i-1}) \tag{5}$$

where $P(w_i \mid w_1,w_2,..,w_{i-1})$ is probability that word w_i follows the word sequence $w_1,w_2,..,w_{i-1}$. Since the weather domain corpus contains a limited amount of sentences a bigram language model is used to approximate P(W). The probability of the word w_i after word w_{i-1} in a bigram language model is calculated by

$$P(w_i \mid w_{i-1}) = \frac{N(w_{i-1}, w_i)}{N(w_{i-1})} \tag{6}$$

where:

$N(w_{i-1},w_i)$ is the frequency of the word pair (w_{i-1},w_i),

$N(w_{i-1})$ is the frequency of the word w_{i-1}.

One major problem with standard N-gram models is that they are estimated from some corpus, and because any particular training corpus is finite, some perfectly acceptable N-grams are bound to be missing from it (Jurafsky & Martin, 2000). To give an example from the domain of speech recognition, if the correct transcription of an utterance contains a bigram $w_{i-1}w_i$ that has never occurred in the training data, we will have $p(w_i \mid w_{i-1})=0$ which will preclude the recognition procedure from selecting the correct word sequence, regardless of how unambiguous the acoustic signal is.

Smoothing is used to address this problem. The term smoothing describes techniques for adjusting the maximum likelihood estimate of probabilities to produce more accurate probabilities. These techniques adjust low probabilities such as zero probabilities upward, and high probabilities downward. Not only do smoothing methods generally prevent zero probabilities, but they also attempt to improve the accuracy of recognition.

Perplexity of the language model represents the branching factor of the number of possible words branching from a previous word. Perplexity PP is defined as:

$$PP = 2^{H(L)} \tag{7}$$

where H(L) represents the entropy of the language and is approximated by:

$$H(L) = -\frac{1}{n}\log_2 P(w_1, w_2, \cdots, w_n) \tag{8}$$

where $P(w_1,w_2,..,w_n)$ is probability of the word sequence $w_1,w_2,..,w_n$, and n is the number of words in a sequence.
In all experiments bigram language model was used. Estimated perplexity of the radio part of the speech database bigram language model is 11.17 for weather domain and 17.16 for the news domain and perplexity of the telephone part of speech database is 17.97.

4. Experiments and results

The word recognition procedure computes the word sequence probability using the Viterbi search in the network of word hidden Markov models and a bigram language model. Word models are constructed from triphone models as shown in figure 4. Additional models for silence, breath noise, paper noise and restarts are used.
All word models are concatenated in parallel and form a single Hidden Markov Model, which is represented by a huge network of nodes. The analysis of an unknown observation sequence is performed by the Viterbi algorithm, producing the maximum a posteriori state sequence of the model with respect to the observed input vectors. Knowing the state sequence of the HMM we can decode the input sequence and transform it into a string of words. Because of the large number of states which have to be considered when computing the Viterbi alignment, a state pruning technique has to be used to reduce the size of the search space. We use the Viterbi beam--search technique which expands the search only to states which probability falls within a specified beam. The probability of reaching a state in the search procedure cannot fall short of the maximum probability by more than a predefined ratio. During the forward search in the HMM N best word sequences are generated using acoustic models and a bigram language model.

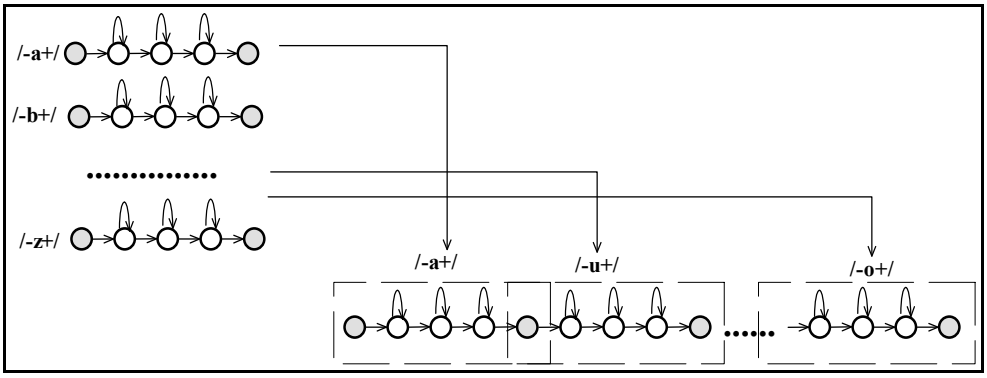


Fig. 4. Word models construction from triphone models.

So far we have performed speech recognition experiments using the radio speech database. The speech database contains weather forecast and news recordings. One part of the database (71%) was used for acoustic modelling and parameter estimation of context dependent phone models, while a smaller part (29%) of the database was used for recognition. All results are given for speaker independent recognition (2 male and 4 female speakers).

Speech recognition results for context-independent and context-dependent speaker independent recognition of the "clean" radio and noisy telephone speech are presented in tables 5 and 6 respectively. Word error rate (WER) results are given for 20 Gaussian mixtures. WER is computed according to:

$$WER = 100\% \left(\frac{W_S + W_D + W_I}{N} \right), \tag{9}$$

where W_S, W_D and W_I are substituted, deleted and inserted words, while N is the total number of words. W_S, W_D and W_I are computed using the Levenshtein distance between the transcribed and recognized sentences.

The increase of the acoustic material in Croatian radio speech recognition resulted with 1.68% decrease of WER. Since the access to the weather information spoken dialog system is planned by telephone, the WER for the telephone data is quite promising. The word error rate for telephone data must be bellow 20% which will be achieved by incorporating more telephone speech in the acoustical model training procedure. And finally both recognition systems performed better when the number of tied states was reduced (using the same phonetic rules) and the number of Gaussian mixtures increased which indicates that more speech should be incorporated in the training of both recognizers for the use in the spoken dialog system.

	RADIO		**TELEPHO.**
	weath. forec.	**news**	**weath. repor.**
Duration [h]	8	13	6
No. words trained	1462	10230	1788
No. words recognized	1462	1462	1788
perplexity	11.17	17.16	17.97
No. Gauss. mix	**% WER**	**%WER**	**%WER**
1	18.7	18.49	30.41
5	13.35	13.13	25.21
10	11.57	11.36	23.18
15	11.11	10.91	22,52
20	**10.54**	**10.58**	**21.76**

Table 5. Croatian speech recognition results: WER computed using monophone HMMs with different number of Gaussian mixtures.

	RADIO		**TELEPHO.**
	weath. forec.	**news**	**weath. repor.**
No. Gauss. mix	**% WER**	**%WER**	**%WER**
1	17.27	14.69	27.16
5	12.76	10.63	21.82
10	11.28	9.56	20.83
15	11.02	9.20	20.49
20	**10.61**	**8.93**	**20.06**

Table 6. Croatian speech recognition results: WER computed using triphone HMMs with different number of Gaussian mixtures.

Graphs in figures 5 and 6 show the word accuracy for monophone and triphone Croatian speech recognition for radio and telephone speech for different numbers of Gaussian mixtures. Word accuracy WA is computed according to:

$$WA = 100\% \left(1 - \frac{W_S + W_D + W_I}{N} \right), \tag{10}$$

The presented recognition results are obtained using 553 tied states for 'clean' radio speech and 377 tied states for telephone speech. Further increase of Gaussian mixture did not increase the accuracy since the speech material is not big enough and a great number of triphones are not present in the training data.

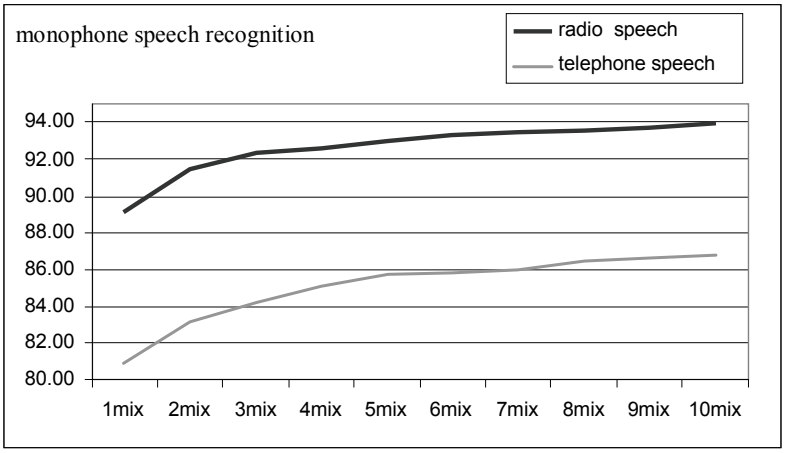


Fig. 5. Word accuracy using monophones for radio and telephone speech.

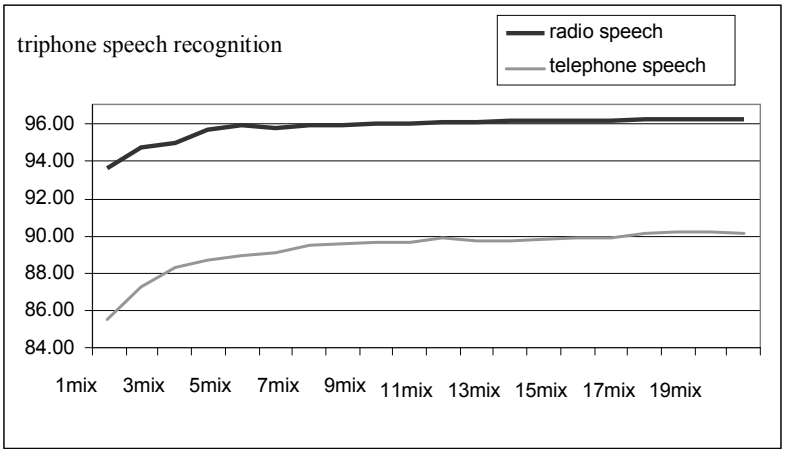


Fig. 6. Word accuracy using triphones for radio and telephone speech.

5. Conclusion

In the paper we described the context-dependent acoustic modelling of Croatian speech in the speech recognition system. An application specific Croatian speech corpus and Croatian phonetic rule were used for context-dependent hidden Markov models based speech recognition. Presented speech recognition system for radio and telephone data is planed for use in the Croatian weather information spoken dialog system.

Speech recognition experiments using context-independent and context-dependent acoustic models were prepared for "clean" radio and for noisy telephone speech. The WER for the radio weather domain is reduced to 10.61% by increasing the number of Gaussian mixtures. The radio speech WER was further reduced to 8.93% by adding the news related speech into acoustical modelling. For the telephone speech 20.06% WER was achieved. The achieved results for telephone speech recognition are promising for further actions in development of the dialog system.

In this work we have shown that the approach for speech recognition using context-dependent acoustical modelling is appropriate for rapid development of limited domain speech applications for low-resourced languages like Croatian. Croatian orthographic-to-phonetic rules are proposed for phonetic dictionary building. The developed Croatian multi-speaker speech corpus was successfully used for development of speech applications. Proposed Croatian phonetic rules captured adequate Croatian phonetic, linguistic and articulatory knowledge for state tying in acoustical models for the speech recognition system. Main advantage of the used approach lies in the fact that speech applications can be efficiently and rapidly ported to other domains of interest under the condition that an adequate speech and language corpus is available.

Since the telephone access to the spoken dialog system is planed, further improvements in speech recognition must be considered. Additionally work on including more speech especially spontaneous speech from different speakers in the corpus is in progress. Further research activities are also planed towards development of the speech understanding module in the dialog system and the speech synthesis module.

6. References

Alumäe, T. and L. Võhandu (2004). Limited-Vocabulary Estonian Continuous Speech Recognition Systems using Hidden Markov Models, *Informatica*, Vol.15(3), 303-314.

Anić, V. and J. Silić (2001). *Pravopis hrvatskoga jezika*, Novi liber. Zagreb. (in Croatian)

Barras, C., Geoffrois, E., Wu, Z. and M. Liberman (2000) Transcriber: use of a tool for assisting speech corpora production. *Speech Communication special issue on Speech Annotation and Corpus Tools.* Vol. 33, No. 1-2.

Black, A., R. Brown, R. Frederking, R. Singh, J. Moody and E. Steinbrecher (2002). TONGUES: Rapid development of a speech–to-speech translation system, Proc. *HLT Workshop*, San Diego, California, pp. 2051-2054.

Duda, R., P. Hart and D. Stork (2001). *Pattern Classification*, John Wiley, Canada, 2001.

Dusan, S. and L. R. Rabiner (2005). On Integrating Insights from Human Speech Perception into Automatic Speech Recognition, *Proc. INTERSPEECH'05-EUROSPEECH,* Lisbon, Portugal, pp. 1233-1236.

Frederking, R., A. Rudnicky and C. Hogan (1997). Interactive Speech Translation in the DIPLOMAT Project, *Proc. Spoken Language Translation Workshop*, Madrid, 61-66.

Furui, S. (2005). 50 Years of Progress in Speech and Speaker Recognition, *Proc. SPCOM'05*, Patras, Grece, 1-9.

Furui, S., M. Nakamura and K. Iwano (2006). Why is Automatic Recognition of Spontaneous Speech So Difficult? *Proc. Large-Scale Knowledge Resources*, Tokyo, Japan, 83-90.

Gauvain, J. L. and L. Lamel (2003). Large Vocabulary Speech Recognition Based on Statistical Methods, in *Pattern Recognition in Speech and Language Processing*, (ed.) Chou, W., (ed.) Juang, B. W., CRC Press LLC, Florida, USA, ch. 5.

Graff, D.(2002) An overview of Broadcast News Corpora. *Speech Communication*, Vol. 37, Issues 1--2, pp. 15-26.

Huang, X. D., A. Acero and H. W. Hon (2000). *Spoken Language Processing: A Guide to theory, Algorithm and System Development*, Prentice Hall, New Jersey, USA.

Hwang, M. Y., X. Huang and F. Alleva (1993). Predicting unseen triphones with senones, Proc. *IEEE ICASSP'*93, 1993, vol. 2, 311-314.

Jelinek, F. (1999). *Statistical Methods for Speech Recognition*, The MIT Press, USA.

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

Kominek, J., Bennett, C. and A. W. Black (2003). Evaluation and correcting phoneme segmentation for unit selection synthesis, *EUROSPEECH ´03*. ISCA. pp. 313-316. Geneva, Switzerland.

Kurimo, M., A. Puurula, E. Arisoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe and M. Saraclar (2006). Unlimited vocabulary speech recognition for agglutinative languages, *ACL HLT Conference,* 487-494. NewYork, USA.

Lee, K., H. Hon and R. Reddy (1990). An Overview of the SPHINX Speech Recognition System, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38(1), 35-45.

Lihan, S., J. Juhar and A. Čižmar (2005). Crosslingual and Bilingual Speech Recognition with Slovak and Czech SpeechDat-E Databases, *Proc. INTERSPEECH'05-EUROSPEECH*, Lisbon, Portugal, 225-228.

Martinčić-Ipšić, S. and I. Ipšić (2004). Recognition of Croatian Broadcast Speech, *Proc. XXVII. MIPRO 2004,* Opatija, Croatia vol. CTS + CIS, p. 111-114.

Martinčić-Ipšić, S. and I. Ipšić (2006a). Croatian Telephone Speech Recognition, *Proc. XXIX. MIPRO 2008,* Opatija, Croatia, vol. CTS + CIS, 182-186.

Odell, J. (1995). The Use of Context in Large Vocabulary Speech Recognition, Ph.D. dissertation, Queen's College, University of Cambridge, Cambridge, UK.

Psutka, J., P. Ircing, J. V. Psutka, V. Radová, W. Byrne, J. Hajič, J. Mírovsky and S. Gustman (2003). Large Vocabulary ASR for Spontaneous Czech in the MALACH Project, *Proc. EUROSPEECH´*03, Geneva, Switzerland, 1821-1824.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE*, vol. 77, no. 2, 257-286.

SAMPA, ESPRIT project1541 Speech Assesment Method, created 1997 on initiative of Bakran and Horga, Phonetics and Linguistics University College London. (accessed May, 2002) `http://www.phon.ucl.ac-uk/hone/sampa/croatian.htm`

Scheytt, P., P. Geutner, A. Waibel (1998). Serbo-Croatian LVCS on the dictation and broadcast news domain, *Proc. IEEE ICASSP'*98, Seattle, Washington.

Schukat-Talamazzini, E. G. (1995). Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algoritmen, Vieweg Verlag, Braunschweig.

Shafran, I. and M. Ostendorf (2003). Acoustic model clustering based on syllable structure, *Computer Speech and Language*, vol. 17, 311-328,

O'Shaughnessy, D. (2003). Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis, *Proc. of IEEE*, 91(9), 1271-1305.

Skripkauskas, M. and L. Telksnys (2006). Automatic Transcription of Lithuanian Text Using Dictionary, *Informatica*, 17(4), 587-600.

Vaičiūnas A. and G. Raškinis (2005). Review of statistical modeling of highly inflected Lithuanian using very large vocabulary, *Proc. INTERSPEECH'05-EUROSPEECH*, Lisbon, Portugal, 1321-1324.

Young, S., G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev and P. Woodland (2002). *The HTK Book*, (for HTK Version 3.2). Cambridge University Engineering Department, Cambridge, UK.

Young, S., J. Odell and P. Woodland (1994). Tree-Based State Tying for High Accuracy Acoustic Modelling, *ARPA HLT Workshop*, Plainsboro, NJ, Morgan Kaufman Publishers, 307-312.

Žibert, J., S. Martinčić-Ipšić, M. Hajdinjak, I. Ipšić and F. Mihelič (2003). Development of a Bilingual Spoken Dialog System for Weather Information Retrieval, *Proc.* EUROSPEECH´03, Geneva, Switzerland, vol. 1, 1917-1920.

Hidden Markov Model Toolkit, Version 3.2, Cambridge University Engineering Department, Cambridge, UK, 2002. http://htk.eng.cam.uk/

9

Speech Technologies for Serbian and Kindred South Slavic Languages

Vlado Delić[1], Milan Sečujski[1], Nikša Jakovljević[1], Marko Janev[1,2], Radovan Obradović[1,2] and Darko Pekar[2]
[1]*Faculty of Technical Sciences, University of Novi Sad,*
[2]*AlfaNum – Speech Technologies, Novi Sad,*
Serbia

1. Introduction

This chapter will present the results of the research and development of speech technologies for Serbian and other kindred South Slavic languages used in five countries of the Western Balkans, carried out by the University of Novi Sad, Serbia in cooperation with the company AlfaNum. The first section will describe particularities of highly inflected languages (such as Serbian and other languages dealt with in this chapter) from the point of view of speech technologies. The following sections will describe the existing speech and language resources for these languages, the automatic speech recognition (ASR) and text-to-speech synthesis (TTS) systems developed on the basis of these resources as well as auxiliary software components designed in order to aid this development. It will be explained how the resources originally built for the Serbian language facilitated the development of speech technologies in Croatian, Bosnian, and Macedonian as well. The paper is concluded by the directions of further research aimed at development of multimodal dialogue systems in South Slavic languages.

1.1 Particularities of highly inflected languages

The complexity of a number of tasks related to natural language processing is directly related to the complexity of the morphology of the language. The principal feature of inflective languages is that words are modified in order to express a wide range of grammatical categories such as tense, person, number, gender and case. Together with a high degree of derivation with the use of prefixes and suffixes typical for such languages, this results in extremely large vocabularies. As a consequence, statistically oriented language models (based on *N*-grams), which are quite successful in modelling languages with a modest degree of morphological complexity, turn out to be inadequate for use for morphologically more complex languages without significant modifications (Jurafsky & Martin, 2000).

The problem affects both automatic speech recognition and text-to-speech synthesis. In the case of ASR, extremely large vocabularies require the existence of extremely large corpora for obtaining robust *N*-gram statistics. For instance, a corpus of English containing 250.000 tokens actually contains approximately 19.000 types (Oravecz & Dienes, 2002), while a

corpus of Serbian of the same size contains approximately 46.000 types (Sečujski, 2009). Furthermore, the rate of out-of-vocabulary (OOV) words is also much higher in case of morphologically rich languages. A number of solutions to this problem have been proposed, mostly based on modelling the statistics of subword units instead of words. Some of the proposed solutions even target South Slavic languages (Sepesy Maučec et al., 2003), however, none of them results in a system of an accuracy sufficient for its practical usability.
The impact of the problem with respect to TTS is related to the difficulty of accurate high-level synthesis. For the text to be delivered to the listener as intelligible and natural-sounding speech, it has to be pre-processed, and most of the activities included require some kind of estimation of robust statistics of the language, as it will be explained in more detail in the following sections. As was the case with ASR, the size of the vocabulary leads to data sparsity, resulting in the need for significantly greater corpora sufficient for obtaining a language model of the same robustness in comparison to languages with a simpler system of morphological categories.
When the four South Slavic languages used in the Western Balkans (namely: Serbian, Croatian, Bosnian, Macedonian) are examined, it can be seen that they exhibit extreme similarities at levels ranging from phonetic and morphological to syntactic and semantic. With the exception of Macedonian, all these languages have until recently been considered as variants of a single language (Serbo-Croatian). Owing to this fact, tools and procedures used for development of most of the resources originally developed for Serbian (including a morphological dictionary (Sečujski, 2002), a morphologically annotated corpus (Sečujski, 2009) and an expert system for part-of-speech tagging (Sečujski, 2005)) were re-used to develop corresponding resources for the other languages. In some cases it was possible to easily create the resources for the other languages by simple modification of existing resources for Serbian, as will be explained in more detail in the following sections.

2. Text-to-Speech

This section will describe AlfaNum TTS, the first fully functional text-to-speech synthesiser in Serbian language, which has been adapted to Croatian, Bosnian and Macedonian as well. It is constantly being improved by introducing novel techniques both at high and low synthesis level (Sečujski et al., 2007).
The high-level synthesis module includes processing of text and its conversion into a suitable data structure describing speech signal to be produced. The output of the high-level synthesis module is a narrow phonetic transcription of the text, containing the information on the string of phonemes/allophones to be produced as well as all relevant prosody information, such as f_0 movement, energy contour and temporal duration of each phonetic segment. The principal modules of a high-level synthesis module are given in Fig. 1.

2.1 High-level synthesis

The text preprocessing module is charged with conversion of text into a format more suitable for text analysis. The text to be preprocessed is usually in a plain format, not even tagged for ends of sentences, and it is up to the *sentence boundary detection module* to locate sentence boundaries, which is the first stage of preprocessing. Most practical systems use heuristic sentence division algorithms for this purpose, and although they can work very well provided enough effort was put in their development, they still suffer from the same

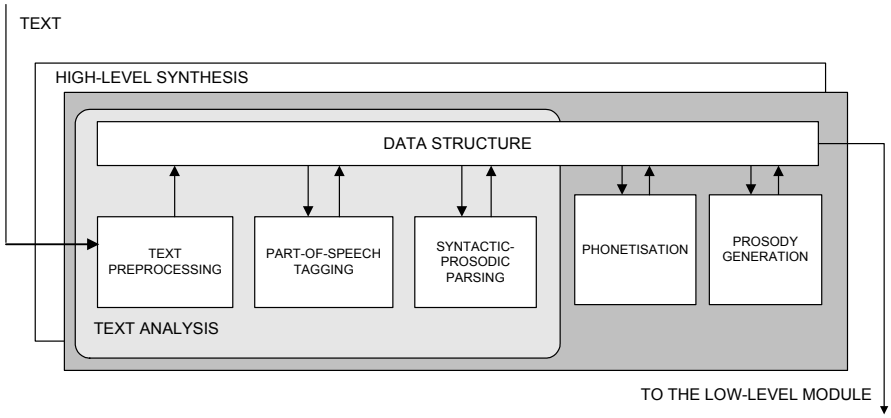


Fig. 1. An overview of the high-level speech synthesis module.

problems of heuristic processes in general – they require a lot of hand-coding and domain knowledge on the part of the person developing the module. Besides neural networks and maximum entropy models, the framework of statistical classification trees can also be effectively used for this purpose, as was first shown in (Riley, 1989). Furthermore, it can be made more powerful by introduction of specialised linguistically motivated features in tree construction. Although the sentence boundary detection module currently used within the AlfaNum TTS system (Sečujski et al., 2002) is a purely heuristic one, development of a tree-based classifier for sentence boundary detection is under way. Further preprocessing stages include conversion of a long string of characters (including whitespaces) into lists of words. Texts, however, do not consist of orthographic words only, and all non-orthographic expressions have to be expanded into words. The preprocessor is thus also charged with processing of punctuation marks, handling acronyms and abbreviations and transcribing numbers into literals. Each of these problems represents a highly language-dependent research area. All of the preprocessing modules currently used by the AlfaNum TTS system for these purposes are of heuristic nature.

Another source of problems is that the surface form of a word is not always a sufficient source of information as to how the word should be read. There is a number of morphological and syntactical ambiguities to be resolved for the word to be read correctly. The critical properties of each word from the point of its conversion into speech are its phonetic transcription as well as the position of accent(s) within it. In the case of all of the aforementioned languages the task of phonetisation is (nearly) trivial, as in each of them one letter basically corresponds to one sounds. The phonology of these languages is rather complex as there are numerous interactions between phonemes at morpheme boundaries, however, almost all of these interactions are reflected in writing as well, and thus do not represent a problem as regards TTS. On the other hand, from the point of view of stress position and type, the situation is less favourable. For example, Serbian, Croatian and Bosnian have an extended system of accentuation, which, from the phonological point of view, has four accents divided into two groups according to their quantity and quality: *long-fall*, *short-fall*, *long-rise* and *short-rise*, their exact realisation varying according to vernacular. Assigning an erroneous accent to a word would affect speech perception to the point that sometimes a completely different meaning would be perceived from the utterance. The

accentuation of Macedonian is somewhat simpler. Besides recent loanwords, word stress in Macedonian is antepenultimate, which means that it falls on the third from last syllable in words with three or more syllables, and on the first syllable in other words. Thus, in most cases, reasonably correct pronunciation of a word does not require its full morpho-syntactic disambiguation.

In general, most of the morpho-syntactic disambiguation required for correct rendering of a word is done through part-of-speech (POS) tagging (although in the case of all of the aforementioned languages there is an occasional dependence of accent type or position on syntax as well). Within the POS tagging procedure, each word has to be assigned some specific additional information related to its morphological status, contained in a unique morphological descriptor or part-of-speech (POS) tag. In case of languages with complex morphology, such tags usually have specified internal structure, and their total number (tagset size) is much larger than in case of languages with simpler morphology (Hajič & Hladká, 1998). This, in turn, leads to the well-known problem of data sparsity, i.e. the fact that the amount of training data necessary increases rapidly with tagset size, making highly accurate part-of-speech taggers for such languages extremely hard to obtain. Whichever of the statistical tagging techniques is used, a number of modifications become necessary when dealing with highly inflective or agglutinative languages (Jurafsky & Martin, 2000). The AlfaNum TTS system performs POS tagging by using a technique that is based on performing a beam-search through a number of partial hypotheses, evaluating them with respect to a database of linguistic rules (Sečujski, 2005). The basic set of rules were hand-coded, however, the database has since been significantly augmented using a transformational-based tagger.

For any partial hypothesis to be considered, the system must know the possible tags for each surface form. However, they cannot be deduced from the surface form itself, which points to the conclusion that any strategy aiming at accurate POS tagging and accent assignment should rely on morphologically oriented dictionaries.

Within this research, by using a software tool created for that purpose, the AlfaNum morphological dictionary of Serbian language was created, containing approximately 100.000 lexemes at this moment, i.e. approximately 3.9 million inflected forms. The research described in this chapter also required that an extensive part-of-speech tagged text corpus be built. Within this research, by using another software tool created for that purpose, the AlfaNum Text Corpus (ATC) was created and part-of-speech tagged, containing approximately 11.000 sentences with approximately 200.000 words in total. Based on the same principles, a Croatian dictionary of approximately the same size was subsequently developed. Owing to extreme similarities of Serbian, Bosnian and Croatian, the Serbian and Croatian dictionaries are jointly used for tagging of Bosnian, and instead of full tagging of Macedonian, only stress assignment is carried out, according to the rule of the antepenultimate syllable and a dictionary of exceptions containing approximately 44.000 types.

Each entry in the AlfaNum morphological dictionary of Serbian, besides the morphological descriptor, also contains the data related to the accentuation of the word, as well as the lemma (base form), which is useful for lemmatisation. The term *entry* thus denotes a particular inflected form of a word, together with the corresponding lemma, values of part-of-speech and morphological categories, as well as its accent structure (a string of characters denoting accent type associated to each syllable). An example of an entry would be:

Vb-p-1-- *uzećemo* (*uzeti*) [\-00].

Morphological categories that are marked are dependent on the part-of-speech, and thus e.g. verbs are marked for tense/mood, gender, number and person, but only in case a particular category is applicable to the tense/mood in question. The example above represents a verb (V) in 1st person (1) plural (p) of the future tense (b), whose surface form is *uzećemo* and whose base form is *uzeti*. The data related to accentuation are given in square brackets. In this way, all the inflected forms of words are present in the dictionary, and the task of part-of-speech tagging of an unknown text amounts (in most cases) to the selection of the correct tag out of all possible tags provided by the dictionary, rather than actual morphological analysis of words.

The dictionary was built in an efficient way using a software tool previously developed for that purpose (Sečujski, 2002). This tool is based on direct implementation of inflectional paradigms of the Serbian language, and its application enables efficient input of complete paradigms instead of individual entries.

When all the possible tags are provided by the dictionary, it remains to select the correct one. As it would be impossible to consider all tag combinations separately, an algorithm similar to dynamic programming is used, keeping the number of partial hypotheses under control.

Let us consider a sentence $W = w_1 w_2 ... w_N$. Each of the words w_i has a corresponding tag list:

$$T_i = \{t_{i1}, t_{i2}, ... t_{iN_i}\}, \tag{1}$$

and its actual tag t_i is one of the t_{ij}, $j = 1, 2, ... N_i$. Initially only the hypotheses of length one are considered, containing only the first word of the sentence:

$$H_1 = \{(t_{11}), (t_{12}), ... (t_{1N_1})\}. \tag{2}$$

In every following step of the algorithm, each variant of the next word is combined with each of the existing partial hypotheses. A set of all possible hypotheses of length two is thus:

$$H_2 = \{(t_{1m}, t_{2n}) \mid m = 1, 2... N_1, n = 1, 2... N_2\}. \tag{3}$$

Each time a new word is appended in such a way, the score of each hypothesis is recalculated, based on the likelihood that a word with such a tag can follow. If the number of all hypotheses exceeds a previously set limit L, only L hypotheses with highest scores are retained, and all the others are discarded. The procedure continues until all words are included and the hypothesis with the highest score is selected as the estimate of actual tag sequence $T = t_1 t_2 ... t_N$. Fig. 2 shows an example of such analysis. The algorithm described here performs in time proportional to the length of the sentence, and one of its interesting features is that it produces partial results very quickly. The first word in the sentence is assigned its tag long before the analysis is over, which is consistent with the notion that, when reading a sentence, humans are usually able to start pronouncing it far before they reach its end, and that they organise the sentence into simple prosodic units which can be obtained from local analysis (Dutoit, 1999). Furthermore, this feature of the algorithm is especially useful from the point of view of speech synthesis, because synthesis of the speech signal can start as soon as the first partial results are obtained, which minimises the delay introduced by POS tagging.

The initial criteria for actual scoring of the hypotheses are based on rules defined according to the statistics of different parts-of-speech in Serbian language and grammatical rules found

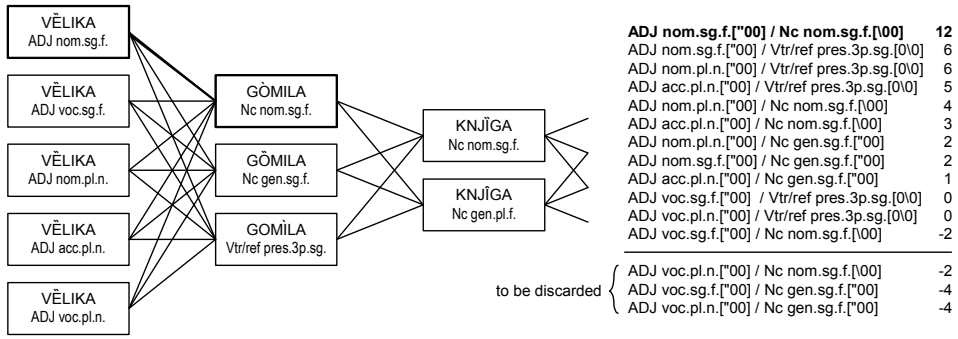


Fig. 2. An example of a step in the disambiguation algorithm for the sentence "*Velika gomila knjiga stoji na stolu*". The diagram shows the situation after all the hypotheses of length two are considered, and three of them with lowest scores are to be discarded (in this example stack size limit is $L = 12$).

in the literature. Further error-correcting rules have been discovered using the transformational-based part-of-speech tagger described in (Sečujski, 2009), and trained on individual sections of the AlfaNum Text Corpus. The tagger is based on the general transformation-based learning paradigm (Brill, 1992), but enhanced with certain learning strategies particularly applicable to highly inflected languages (Sečujski, 2009). Both hand-coded and automatically obtained rules are created following standard templates such as:
Award n points to a partial hypothesis $h = (w_1, w_2, \ldots w_l)$:

- If w_l is tagged t_i
- If w_l is tagged t_i and w_{l-1} is tagged t_j
- If w_k is tagged t_i, w_{l-1} is tagged t_j and w_{l-2} is tagged t_k
- If w_l is tagged t_i and w_{l-1} is tagged t_j and the value of a morphologic category c contained in the tag t_i is the same (is not the same) as the value of the corresponding morphologic category contained in the tag t_j
- If w_l is tagged t_i and w_{l-1} is tagged t_j and all of the values of morphologic categories c_1, c_2,...c_k contained in the tag t_i are the same (are not the same) as the values of corresponding morphologic categories contained in the tag t_j

where n is assigned depending on the technique used.
After the (presumably) correct tag sequence has been discovered, the next step consists of modifying accent patterns to account for occasional dependence of accent type and/or position on syntax, as described previously, and performing syntactic-prosodic parsing of the sentence (detecting prosodic events such as major and minor phrase breaks, setting sentence focus etc.). Both are currently done using heuristic algorithms, however, the development of a tree-based classifier which would be in charge of the latter is under way. This classifier will be trained on sections of the AlfaNum Text Corpus which are annotated for minor and major phrase breaks as well as sentence focus.
It remains to assign each word its actual prosodic features, such as durations of each phonetic segment as well as f_0 and energy contours. In the version for the Serbian language, this is currently performed using regression trees trained on the same speech database used for speech synthesis. The section of the database used for training of regression trees is fully annotated with phone and word boundaries, positions of particular accent types and pro-

sodic events such as major and minor phrase breaks and sentence focus. Separate regression trees are used for prediction of phonetic durations and for prediction of f_0 and energy contours. Owing to this approach, actual acoustic realisation of each accent in synthesised speech is expected to correspond to the most common realisation of the same accent in a phonetically and prosodically similar context in the speech database. The listening experiments carried out so far have confirmed the expectation that such an approach would lead to superior naturalness of synthetic speech in comparison with the previous version, which was based on heuristic assignment of predefined f_0 and energy contours corresponding to particular accentuation configurations (Sečujski et al., 2002). The versions of the synthesiser for Croatian, Bosnian and Macedonian language still use the heuristic algorithm for prosody prediction, however, the Croatian synthesiser is expected to switch to regression-tree based prosody prediction soon, as prosodic annotation of the Croatian speech database is currently under way. As was the case with morphological dictionaries, significant experience in creation of other resources for the Serbian language will certainly contribute to efficient creation of appropriate resources for other kindred languages as well.

2.2 Low-level synthesis

The term low-level synthesis refers to the actual process of producing a sound that is supposed to imitate human speech as closely as possible, based on the output of the high-level synthesis module described in the previous subsection. In all of the available versions of the system, the concatenative approach has been used as being the simplest and at the same time offering high intelligibility and reasonably high flexibility in modifying prosodic features of available phonetic segments prior to synthesis (Sečujski et al., 2002).

The AlfaNum R&D team has recently recorded a new speech database containing 10 hours of speech from a single speaker (instead of a 2.5 hour database previously used), and so far annotated approximately 3 hours of it using visual software tools specially designed for that purpose (Obradović & Pekar, 2000). By keeping score of the identity of each phone in the database and its relevant characteristics (such as the quality of articulation, nasalisation and vocal fry), use of phones in less than appropriate contexts was discouraged, which further contributed to overall synthesised speech quality. Unlike most other synthesisers developed for kindred languages so far, the AlfaNum TTS engine can use larger speech segments from the database, according to both phonetic and prosodic requirements, and select them at runtime in order to produce the most intelligible and natural-sounding utterance for a given plain text (Beutnagel et al., 1999). The full increase in synthesis quality is yet to come after the remaining 7 hours of speech are annotated.

According to differences between the existing and the required values of parameters previously defined, each speech segment which can be extracted and used for synthesis is assigned *target cost*, and according to differences at the boundaries between two segments, each pair of segments which can be concatenated is assigned *concatenation cost*. Target cost is the measure of dissimilarity between existing and required prosodic features of segments, including duration, f_0, energy and spectral mismatch. Concatenation cost is the measure of mismatch of the same features across unit boundaries. The degree of impairment of phones is also taken into account when selecting segments, as explained previously. The task of the synthesiser is to find a best path through a trellis which represents the sentence, that is, the path along which the least overall cost is accumulated. The chosen path determines which segments are to be used for concatenation, as shown in Fig. 3, with s_{ij} denoting segments, c'_{ij}

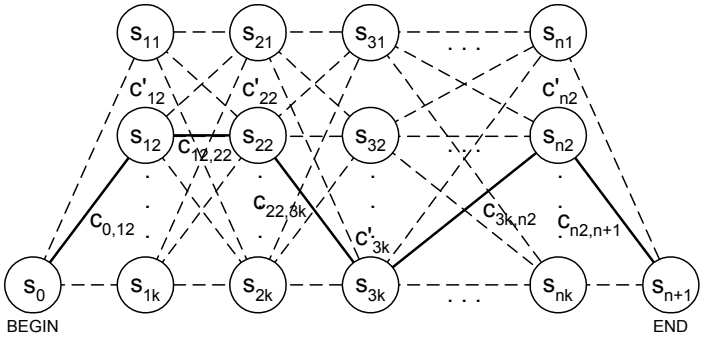


Fig. 3. Finding the best path through a trellis representing a sentence.

denoting segment costs and $c_{ij,pq}$ denoting concatenation costs. Segment modifications related to smoothing and prosody manipulation are carried out using the TD-PSOLA algorithm.

In a version which is currently under development, an alternative to the TD-PSOLA low-level synthesis algorithm is being introduced – HMM based synthesis (Tokuda et al., 2000). Segmental intelligibility tests have still to be carried out, yet the first results seem to be encouraging.

3. Automatic speech recognition

AlfaNum automatic speech recognition (ASR) system as well as most of state-of-the-art systems is based on hidden Markov models (HMM). State emitting probabilities are modelled by Gaussian mixture models (GMM), with each Gaussian distribution defined by its mean and full covariance matrix. The parameters of each Gaussian in GMM are estimated using the Quadratic Bayesian classifier (Webb, 1999), which is a generalisation of the standard K-means classification iterative procedure. The goal of decoding in the AlfaNum ASR systems is to find the most probable word sequence corresponding to the input speech, as well as a confidence measure for each recognition. Viterbi algorithm is used for a search for the most probable word sequence. To accelerate the search procedure, beam search and Gaussian selection (Janev et al., 2008) are used.

3.1 Speech corpus

One of the first steps in development of an ASR system is speech corpus acquisition. Since 1998 a speech corpus has been developed for Serbian according to the SpeechDat(E) standard (Delić, 2000). It contains utterances from about 800 native speakers (400 male and 400 female), which have been recorded via the public switched telephone network. Today, the corpus volume is about 12 hours of speech (silent and damaged segments are excluded). A section of the corpus, containing 30 minutes of speech from about 180 speakers (100 male and 80 female), is used as the test set for the experiments. Transcriptions are at the phone level, and boundaries between phones are corrected manually (Obradović & Pekar, 2000). The language of the speech corpus is Serbian, but it is used for development of ASR applications in Croatian and Bosnian as well, since the phonetic inventories of these kindred languages are practically identical, with minor variations in pronunciation of certain phonemes.

3.2 Acoustic models

For the purposes of ASR, several changes had to be introduced into the phonetic inventory of the Serbian language. Instead of the standard 5 vowels in Serbian i.e. /i/, /e/, /a/, /o/ and /u/ (IPA notation), two sets containing 5 long and 5 short vowels are taken into consideration. This distinction has been motivated by the fact that short vowels usually do not reach its target position. A vowel is marked as long, if its duration is longer than 75 ms and its average energy is greater than 94% of average vowel energy in the utterance containing the vowel, otherwise the vowel is marked as short. Phone /ə/ is regarded as a standard vowel as well. Moreover, closure and explosion (friction) of stops (affricates) are modelled separately in order to obtain more precise initial models. These models will be referred as sub-phones in further text.

Acoustic features of phone are influenced by articulatory properties of nearby phones, and this influence is called coarticulation. In order to capture acoustic variations of phone caused by coarticulation, triphone (context dependent phone/sub-phone) is used as basic modelling unit (Young et. al., 1994). Introducing sub-phone models results in the slightly complex procedure for conversion of words into appropriate sequence of triphones, where sub-phone models are treated as a single phone. Silence and non-speech sounds (various types of impulse noise) are modelled as context independent units.

The number of HMM states per model is proportional to the average duration of all the instances of the corresponding phone in the training database (e.g. long vowels are modelled by five states and stop explosions by only one state). On this way slightly better modelling of path in feature space is achieved at the cost of reducing the number of observations per state.

The number of mixtures per HMM state is determined semi-automatically. It gradually increases until the average log likelihood on the validation set starts to decrease or the maximum number of mixtures for the given state is reached. Maximum number of mixtures per state depends on which model that state belongs. For example, models for fricatives /s/ and /ʃ/ have fewer mixtures per state than vowels, because the coarticulation effects on these fricatives are smaller than on vowels.

Using triphones instead of monophones leads to a very large set of models and insufficient training data for each triphone. All HMM state distributions would be robustly estimated if sufficient observations were available for each state. This could be achieved by extending the training corpus or by including observations related to acoustically similar states. The second solution, known as tying procedure, was chosen as being less expensive, even though it generates some suboptimal models.

3.3 Tying procedure

The main issue in the tying procedure is how to define acoustically similar states. The vocal articulators are moved at relatively slow speeds and do not remain in the steady positions through the duration of a phone. They are moving from the position required to articulate the preceding phone to the position required for the successive phone, via the position needed for the current phone. Therefore, acoustically similar states are the states of the same phone at the same position in HMM (left-to-right model topology is used), which have phones with a similar place and manner of articulation in their context. The level of the state similarity depends on the similarity of its contexts. The previous phone has more influence on the initial HMM states than on the final HMM states, and subsequent phone has more

influence on the final states than on the initial. Hence the position of the state in HMM defines the importance of the context. For the initial state, and all states close to it, the left context is more important, and for the final state, all states close to it and central state (if such a state exists) the right context is more important (See example in Fig. 4.). It is obvious that the states with the same more important context and a different less important context are more similar than vice versa (Young et. al., 1994).

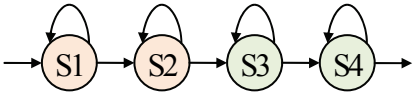


Fig. 4. Left-to-right HMM topology with 4 emitting states. Important context for states 1 and 2 is left, and for states 3 and 4 is right context.

For the tying procedure, it is necessary to define phone similarity. Definition of phone similarity is based on our linguistic knowledge about the place and manner of articulation of the phone. Fig. 5. illustrates similarity level tree. IPA notation is used for the phone labels. Non-speech sounds like silence, background noise and unarticulated sounds are marked by 'sil', 'int' and 'unk', respectively.

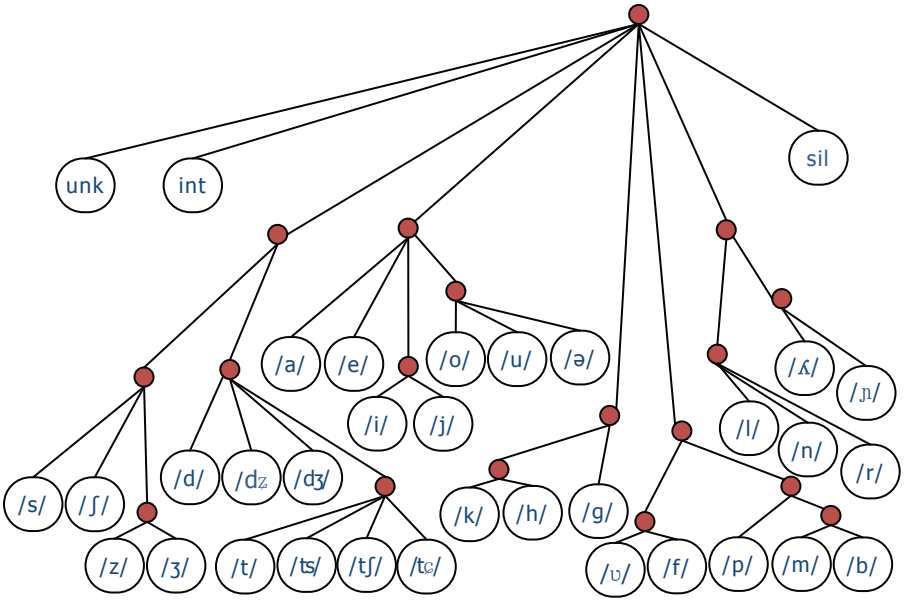


Fig. 5. The tree of the phonetic similarity. Closure and explosion (friction) of stops (affricates) are treated as single context.

The tying procedure (Fig. 6) is applied only to the states with an insufficient number of observations. Mark with *Si* the *i*-th HMM state of the phone *Ph* (*i* is the indicator of state position in left-to-right HMM topology as well). The more important context for the state *Si* is *MIC* and less important context is *LIC*. Suppose that *Si* has an insufficient number of observations for robust parameter estimation. The proposed algorithm attempts to obtain the additional observations for the state *Si*, by borrowing observations from the *i*-th states,

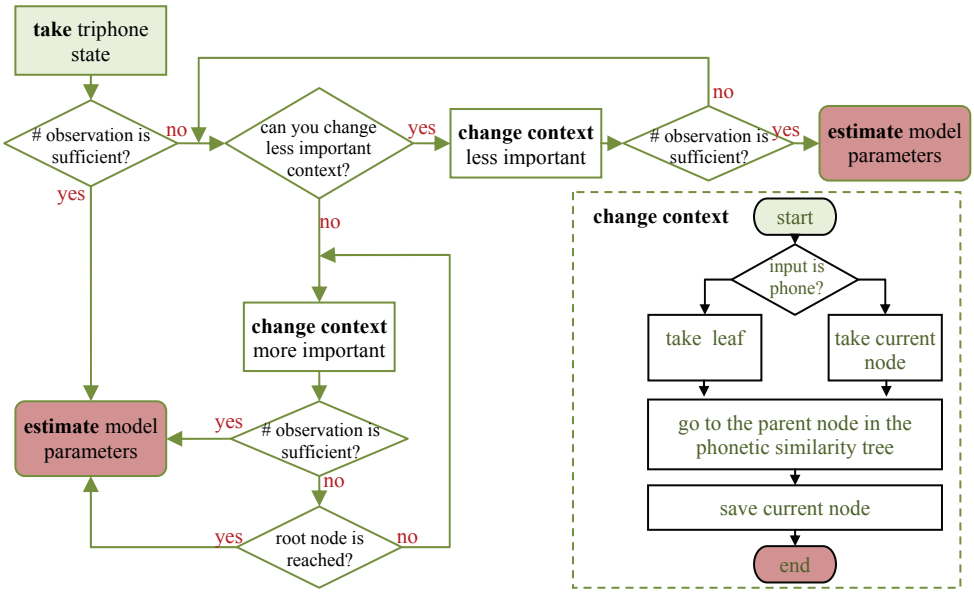


Fig. 6. Flowchart of the tying procedure.

modelling the phone *Ph* being in different contexts. The algorithm starts with the states whose more important context is *MIC* and the less important context is any phone in parent node of the phonetic similarity tree for the phone *LIC*. If in this attempt the sufficient number of observations is not obtained, the algorithm extends the search to states belonging to the *i*-th state of the phone *Ph* whose more important context is *MIC* and less important context is any phone contained by one step higher parent node containing the phone *LIC*. The previous step is repeated until the sufficient number of observations is obtained or the root node is reached. If the root node is reached and a sufficient number of observations is not, then the algorithm tries to borrow additional observations from the *i*-th state of the phone *Ph*, whose the less important context is arbitrary and the more important context is any phone in the parent node containing phone *MIC*. If in this attempt a sufficient number of observations is not obtained, the algorithm extends the search on states, which belong to the *i*-th state of the phone *Ph* whose less important context is arbitrary and more important context is any phone in the one step higher parent node containing phone *MIC*. The previous step is repeated until a sufficient number of observations is obtained or the root node is reached (Delić at al., 2007).

3.4 Vocal tract length normalisation

Acoustic variations between training and test conditions, caused by different microphones, channels, background noise as well as speakers, are known to deteriorate ASR performance. Variations caused by speakers can be divided into extrinsic and intrinsic. Extrinsic variations are related to cultural variations among speakers as well as their emotional state, resulting in diverse speech prosody features. Intrinsic variations are related to speaker anatomy (vocal tract dimensions).

The state-of-the-art ASR systems based on HMM and GMM are sensitive to differences in training and test conditions, which result in serious degradations of performance (Molau, 2003; Benzeghiba et al., 2006). One of the common methods to reduce spectral variations caused by different vocal tract length and shape is vocal tract length normalisation (VTN). There are several algorithms proposed in the literature. There are two approaches based on: *i*) formant position and *ii*) maximum likelihood criterion. The goal of the algorithms based on formant position is to find spectrum frequency warping function which map average (sample mean or median) formant position of some speaker into average formant position of universal speaker (Gouvea & Stern, 1997; Jakovljević et al., 2006). On the other hand, the goal of the algorithms based on the maximum likelihood criterion is to find spectrum frequency warping function, which transforms feature vectors of some speaker on the way which leads to increased theirs likelihood on the universal speaker model (Lee & Rose, 1996; Welling at al., 1999). Modification of this approach is presented in (Miguel et al., 2008) where this transformation is incorporated into a so called 2-D HMM model.
The work presented in this chapter is based on (Welling et al., 1999). Piecewise linear spectrum warping function is chosen as the most effective one and its implementation the simplest one.
It is defined as:

$$\omega_\alpha = \begin{cases} \alpha\omega & \omega \le 7\pi/8 \\ \alpha\omega - (8-7\alpha)(\omega - 7\pi/8) & 7\pi/8 \le \omega \le \pi \end{cases} \tag{4}$$

where ω is the original frequency and ω_α scaled frequency and α VTN coefficient. In order to reduce search space, VTN coefficients are discrete and usually take values from 0.88 up to 1.12 with step 0.02.
The criterion to choose VTN coefficient is:

$$\alpha_r = \arg\max_\alpha P(X_{r,\alpha} \mid W_r; \lambda_k) \tag{5}$$

where $X_{r,\alpha}$ are all feature vectors which belong to the speaker r normalised by the VTN coefficient α, and W_r are the corresponding transcriptions, and λ_k model of the universal speaker.
The training procedure can be summarised into two steps:

1. VTN coefficient estimation for each speaker in the training phase;
2. Training of HMM models which will be used in the recognition process.

Additionally, the test procedure basing on a multiple pass strategy includes three steps:

1. Initial recognition of the original (unnormalised) sequence of the feature vectors using a speaker independent model set. The output consists of initial transcription and phoneme boundaries;
2. VTN estimation using initial transcription generated in the previous step. The procedures of VTN coefficient estimation are the same as those in the training process. Note that estimation of VTN coefficients in the test procedure is burdened with additional uncertainty because initial transcriptions and phone boundaries can be incorrect (which is not the case in the training phase);
3. Final recognition of the sequence of feature vectors normalised by the VTN coefficient estimated in the previous step. The VTN coefficients are estimated by using a speaker independent ASR system trained on the normalised features.

The models with one Gaussian per HMM state are chosen as models for VTN estimation, because of their general nature and the fact that they do not adapt to the features of a particular speaker, unlike HMM models with more than one Gaussian mixture per state (Welling et al., 1999).

We claim that the disadvantage of the standard procedure for VTN coefficient estimation defined by (5) is it's favouring of longer and more frequent phonemes (their frames are dominant in likelihood estimation of the sequence). Here we suggest several optional criteria. For the sake of convenience the method described by (5) in the further text will be referred to as M0.

In order to eliminate the influence of phone duration on VTN coefficient estimation, the value which maximises average likelihood per phone instance should be used as VTN coefficient. The term "phone instance" stands for one particular realisation of corresponding phoneme in the speech corpus. This criterion can be summarised as:

$$\alpha_r = \arg\max_{\alpha} \frac{1}{N_{pi}} \sum_{n=1}^{N_{pi}} P_n(X_{n,r,\alpha} \mid W_n; \lambda_k) \tag{6}$$

where $P_n(X_{n,r,a} \mid W_n; \lambda_k)$ is the likelihood of the phone instance W_n on the universal model set λ_k and the observations belonging to the given phone instance $X_{n,r,a}$, N_{pi} is the number of the all phone instances belonging to the speaker r. The scaling factor $1/N_{pi}$ is not essential, but for comparison of the average values between different speakers it is. The likelihood of the phone instance can be calculated as sample mean or sample median of the likelihoods of the observations belonging to the phone instance. The first variant in the further text will be referred to as M1 and the second as M2. Favouring phonemes with more instances in the corpus was motivated by the idea to choose a VTN which results in higher likelihood for a larger number of phone instances, and in vowels as most frequent phonemes. The weakness of this method is that it does not result in the optimal increase of word sequence likelihood, since phone instances of longer durations have greater influence than phone instances of shorter durations. Note that the goal of training and test (decoding) procedure is to obtain the maximum likelihood of word sequence. The motivation for M2 method is similar to the one for the M1 method, with an additional aim of experimenting with robust methods for estimation of likelihood of phone instances. With the use of sample median instead of sample mean the influence of extremely low and high values of feature vector likelihood is eliminated.

In order to eliminate the influence of phone duration and frequency in VTN coefficient estimation, the value which maximises average likelihood per phoneme should be used as the VTN coefficient. The likelihood per phoneme represents the average of the likelihoods of all feature vectors belonging to the given phoneme. We proposed four variants which differ in the way how average likelihood per phone and average phone likelihood is calculated. The method, which is in further text referred to as M3, calculates both average likelihood per phoneme and average phoneme likelihood as sample mean. The method referred as M4 is similar to the M3, but it calculates average phoneme likelihood as sample median. The methods referred to as M5 and M6 are similar to the M3 and M4 respectively, but they calculate average likelihood per phoneme as sample median.

None of the methods M3-M6 results in the increase of the likelihood of word sequence. The M4 method represents a robust version of the M3 method. The explanation is the same as

the one for the M2 method. The M5 and M6 methods represent robust versions of the M3 and M4 methods respectively. The use of sample median instead of sample mean results in the elimination of influence of extremely low and high values of phoneme likelihoods.
None of the proposed methods take into consideration non-speech, damaged segments and segments with occlusions of plosives and affricates. All of them use the same initial model set (with one Gaussian per state). All final model sets have the same topology i.e. the number of models, states and mixtures.
The standard features used in VTN estimation procedure are the same as the features used in the recognition process. This approach is based on the reasoning that a VTN coefficient should reduce inter-cluster variations for both static and dynamic features, although the theoretical motivation for VTN includes only spectrum envelope modifications (static features).
However, in the histogram which represents the frequency of the VTN coefficients in the training corpus, there is a significant peak at 1.04 for the female speakers, as shown in Fig. 7. The analysis of the causes which lead to the peak at 1.04 in the histogram included the analysis of the curves describing the dependency of average likelihood on VTN coefficients. These are the curves used for VTN estimation (the estimated value of a VTN coefficient is the point where the curve reaches its maximum). These curves for a majority of the female speakers with estimated VTN value equal to 1.04, are bimodal (two close local maxima, as shown in Fig. 8. a)) instead of unimodal (only one local maximum, as shown in Fig. 8. b)), the latter being expected as more common.
Excluding dynamic features from the VTN estimation procedure results in a unimodal shape of the decision curves for all speakers. The values of word error rate WER on the standard test corpus for all estimation methods are presented in Table 1. The cases when only static and both static and dynamic features are used are given in the first and second row, respectively. The results show that if dynamic features are omitted, the WER is smaller for a majority of the proposed methods of VTN estimation. In the case of M6 method, the opposite result is caused by smaller efficiency of the sample median in the test phase. The same holds for the M5 method, but the result was not contrary to the majority.

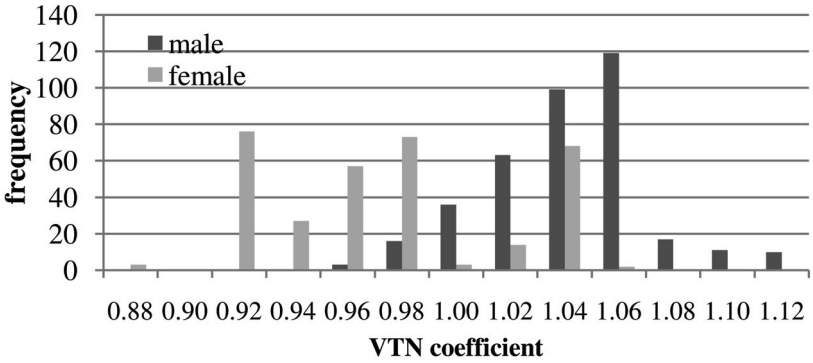
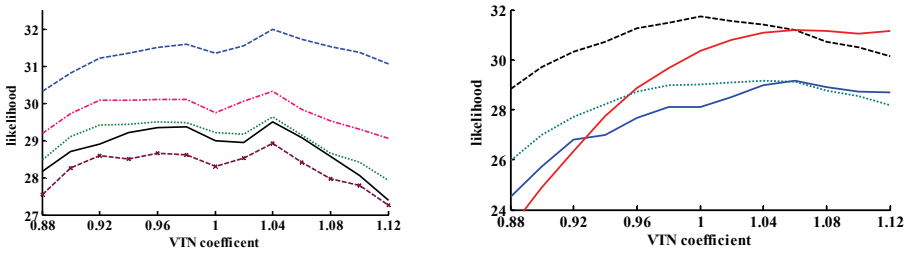


Fig. 7. The histogram of VTN coefficients for male and female speakers in the training corpus in case of M0 estimation method. For other proposed methods similar histograms are obtained.



a) b)

Fig. 8. a) The examples of the bimodal shapes of the VTN decision curves typical for the most female speakers with VTN coefficient equal to 1.04. b) The examples of unimodal shapes of the VTN decision curves typical for the majority of the speakers.

	M0	M1	M2	M3	M4	M5	M6
s	4.28	4.52	4.38	4.07	4.38	4.38	4.59
s+d	4.45	4.66	4.80	4.66	4.38	4.90	4.49

Table 1. The values of WER for the methods of VTN estimation depending on whether static or both static and dynamic features are used

	M0	M1	M2	M3	M4	M5	M6
norm.	4.28	4.52	4.38	4.07	4.38	4.38	4.59
unnorm.	5.07	5.31	5.11	4.76	4.55	5.42	4.61

Table 2. The values of WER for the methods of VTN estimation in case the HMM set is trained on normalised (norm.) or unnormalised (unnorm.) features

The motivation to explore the necessity for the iterative VTN coefficient estimation in the training phase is based on the fact that initial results showed significant differences depending on whether an HMM set, used for the VTN estimation, was trained on the normalised or on the unnormalised set of features. The results are shown in Table 2. Note that both HMM sets used in the VTN estimation procedure have the same complexity i.e. they consist of a single Gaussian density per triphone state. These differences suggest that VTN values estimated in the training phase could be improved (so as to result in a lower WER), suggesting that an iterative procedure should be adopted.

The iterative procedure can be summarised into the following three steps:

1. An HMM set λ_k, in the k-th iteration step, containing triphone states with a single Gaussian density, is trained on the feature vectors normalised by appropriate VTN coefficients for each speaker. The VTN coefficient values are in the initial step equal to 1 for all speakers and in the other steps equal to the values estimated in the previous step.
2. For each speaker in r the training corpus, a VTN coefficient a_r is chosen as the value which maximises the average likelihood per observation or phone instance or phoneme depending of method (M0-M6).
3. Repeat steps 1 and 2 until the number of changes or average change becomes sufficiently small. In this paper, the stopping condition is satisfied when the average change of VTN coefficients becomes smaller than one half of the VTN coefficient step (i.e. 0.01).

	#sub	#ins	#del	WER[%]	RI1[%]	RI2[%]	RI3[%]
REF1	94	56	9	5.94			
REF2	94	51	8	5.28			
M0	65	44	6	3.97	27.7	24.8	11.1
M1	65	39	5	3.76	31.4	28.7	16.8
M2	60	36	5	3.49	36.5	34.0	20.3
M3	69	42	8	4.11	25.2	22.2	10.0
M4	64	46	7	4.04	26.4	23.5	11.1
M5	66	39	5	3.80	30.8	28.1	16.0
M6	69	50	9	4.42	19.5	16.3	2.2

Table 3. Performance of the analysed system and its relative improvement in comparison to three referent systems (REF1, REF2, original M0). The method M0, whose performance is shown in the table, is different from the original M0 in that it uses only static features and iterative procedure for VTN estimation.

The complete results are presented in Table 3. The first referent system (REF1) represents a speaker independent ASR system. The complexity of this system is the same as the complexity of all systems which used VTN. The second referent system (REF2) is a gender dependent ASR system, with slightly smaller complexity than the other ASR systems which are analysed. The remaining systems include VTN estimation, differing between themselves in the type of VTN estimation used. Their relative improvements (RI) in comparison to REF1, REF2 and basic M0 method proposed in (Welling et al., 1999) are presented in the last three columns of Table 3, respectively.
All VTN system results in significant RI comparing to the referent systems REF1 and REF2. VTN methods M1 and M2 achieve the best performance, but McNemar test (Gillick & Cox, 1989) shows that the differences are not statistically significant in comparison to the method M0 (only static features and iterative VTN estimation procedure), M4 and M5.
Some of the proposed VTN estimation methods results in noteworthy RI comparing to baseline VTN methods (see RI3 for M1 and M2). These differences are proved statistically significant by McNemar test. A possible explanation could be that vowels are frequent phonemes and they contain more information about vocal tract length then other phonemes. The VTN estimation methods which disregard frequency and duration of phonemes (M3-M6) demonstrate significant variations in WER depending on whether the sample mean or the median is used. These variations are probably the result of an insufficient number of instances in the test phase. The results of the experiments with fast VTN tests support the previous statement (Jakovljević, 2009). The improvement in the case of M4 and M6 is minor, which can be explained by small efficiency of sample median used for estimation of average phone likelihood on the test set.

3.5 Gaussian selection

In order to obtain a high level of accuracy, HMM based CSR systems typically use continuous densities. Most of them tend to operate several times slower than real time which eventually makes them too slow for any real–time application. In such systems, calculation of state likelihoods makes up a significant proportion (between 30-70%) of the

computational load. Actually, each state usually contains a significant number of Gaussian components in the corresponding mixture that are all separately evaluated in order to determine the overall state likelihood. Many techniques could be applied in order to reduce the computations required. Some of them target dimensionality reduction (like linear discriminant analysis or heteroscedastic linear discriminant analysis), some of them tying of acoustical states (semi-continuous HMM models), and there is also a number of fast Gaussian Selection (GS) methods that for each frame obtain the desired set of baseline Gaussians to be calculated exactly, based on a pre defined data structure. Of course, the goal is to increase the speed of speech recognition system without degrading the recognition accuracy. There are two distinct classes of GS methods: bucket box intersection (Woszczyna et al., 1997) and clustering (Bocchieri, 1993), (Knill et al., 1996), (Knill et al., 1999). We developed our own GS method, which is described in detail in (Janev et al., 2008).

The basic idea behind the clustering GS method is to form hyper-mixtures by clustering close baseline Gaussian components into a single group (clusters) by means of Vector Quantisation (VQ) assigning to each cluster unique hyper-density (almost always Gaussian) with parameters estimated in the appropriate way. In the decoding process, only those baseline Gaussian components belonging to clusters with corresponding hyper-densities whose "distance" to the particular speech frame is above predefined threshold are calculated directly, while the likelihood of others are floored with some approximate values. It significantly improves computational efficiency with relatively small degradation in recognition performances (Janev et al., 2008). There is no problem if the overlaps between Gaussian components are small, and their variances are of the same range. However, in real case, there are numerous models which do not fit this profile. Actually, significant overlapping between Gaussian components is common situation in CSR systems.

Baseline VQ based Gaussian selection is based on (Bocchieri, 1993). Actually, during the training phase the acoustical space is divided up into a set of VQ regions. Each Gaussian component (mixture) is then assigned to one or more VQ codewords (VQ Gaussian mixture clustering). During the recognition phase, the input feature vector is vector quantised, i.e. the vector is mapped to a single VQ codeword. The likelihood of each Gaussian component in this codeword shortlist is computed exactly, whereas for the remaining Gaussian components the likelihood is floored i.e. approximated with some back-off value. The clustering divergence that we have used in VQ based approach was of course different than the one that used in (Bocchieri, 1993) because it is not suitable enough for application with full covariance Gaussians. It was taken from the more theoretical works presented in (Goldberg et al., 2005) and (Banerjee et al., 2005). It is the most appropriate and theoretically motivated approach for the simplification of a large Gaussian mixture (with large number of components) into smaller (Shinoda et al., 2001), (Simonin et al., 1998), which is a significant part of the problem in the GS clustering approach. It can be showed that generalised k-means clustering leads to the local minimum of the target function that represents symmetric KL divergence between the baseline Gaussian mixture f and its simplification g:

$$D(f||g) = \sum_{i=1}^{k} \alpha_i \min_{j=1}^{n} KL(f_i||g_j), \tag{7}$$

where f_i and g_j are components of mixtures f and g, and a_i is the occupance of f_i. This is actually a generalisation of the well known Lindo-Buzo-Gray algorithm (Knill et al., 1996), (Lindo et al., 1995). The algorithm actually obtains the local minimum of $D(f||g)$ by

iteratively repeating REGROUP and REFIT steps. In the REGROUP step, every baseline Gaussian component θ_m is assigned to the unique cluster chosen so that the symmetric KL divergence $KL(\theta_m, \theta_f)$ to the hyper-Gaussian θ_f that corresponds to cluster is minimal. In the REFIT step, parameters of the "new" hyper-Gaussian (c_f, Σ_f) that correspond to the particular cluster are estimated in the Maximum Likelihood manner i.e. equivalently as the ones that minimise the KL divergence between the underlying Gaussian mixture that corresponds to the particular cluster and the actual hyper-Gaussian (Banerjee et al., 2005):

$$\hat{c}_f = \sum_{m=1}^{M_f} w_m \mu_m \tag{8}$$

$$\hat{\Sigma}_f = W_f + \sum_{m=1}^{M_f} w_m (\hat{\mu}_m - \hat{c}_f)(\hat{\mu}_m - \hat{c}_f)^T \tag{9}$$

$$W_f = \sum_{m=1}^{M_f} w_m \hat{\Sigma}_m \tag{10}$$

The term W_f is the pool covariance matrix of the f-th cluster, while w_m is the mixture cluster occupancy (the whole concept could be given strait forward in the terms of soft posterior probabilities obtained using Baum Welch algorithm, but are omitted for the simplicity as in (Janev et al., 2008)).

The main idea how to decrease the influence of significant overlapping of baseline Gaussians is for GS process to be driven by the eigenvalues of covariance matrices of Gaussians to be selected. The basic idea is to group the baseline Gaussian components on the basis of their eigenvalues into several groups, before the actual VQ clustering is applied on each group separately. The method is referred as Eigenvalues Driven Gaussian Selection (EDGS). If the baseline VQ clustering is performed on the whole set of Gaussian components, then at the end of the procedure, in some cluster, there could be both components for which the eigenvalues of covariance matrices are predominantly large, and those for which the eigenvalues of covariance matrices are predominantly small. This is especially the case if the degree of Gaussian components overlapping is high, because many low-variance mixtures could be masked by high-variance ones and thus assigned to the same cluster. This comes as a consequence of the use of symmetric KL clustering distance, more precisely, its Mahalanobis component. As a result, the covariance matrix of the hyper-Gaussian that corresponds to a cluster can have predominantly large eigenvalues, although there are many baseline Gaussian components belonging to that cluster with predominantly small eigenvalues of covariance matrices.

Baseline Gaussian components are masked by high-variance ("wide") ones, thus in the decoding process the following can happen. If the likelihood of a hyper-Gaussian evaluated on the input vector is above the predefined threshold, all baseline components in the cluster will be evaluated for that particular input vector.

The performance of a Gaussian selection procedure is assessed in terms of both recognition performance and reduction in the number of Gaussian components calculated. Reduction is described by the computation fraction CF, given as CF = $(G_{new} + R_{comp})/G_{full}$, where G_{new} and G_{full} are the average number of Gaussians calculated per frame in the VQGS and the full system respectively, and R_{comp} is the number of computations required for the system to

calculate log-likelihoods of hyper-mixtures in order to decide whether the mixtures belonging to that cluster will be evaluated or not. The evaluation will include even those mixtures with low likelihood values that should have been excluded from the evaluation in order to obtain a sufficient reduction in computational load and at the same time not to change WER significantly. The result is the increase in both CF and WER. It is essentially for EDGS to work that we keep the average number of baseline components in cluster n_{avr} reasonably small. Nevertheless, the similar constraint must also be met in order to obtain satisfactory recognition accuracy of any GS system.

As a result of situations when low-variance ("narrow") components are masked by high-variance ("wide") ones, in the decoding process the following can happen. If the likelihood of a hyper-Gaussian evaluated on the input vector is above the predefined threshold, all the baseline components in the cluster will be evaluated for that particular input vector. The evaluation will include even those components with low likelihood values that should have been excluded from the evaluation in order to obtain a sufficiently low CF and at the same time not to change WER significantly. The result is the increase in both CF and WER. Thus, EDGS proceeds with the combining of the most significant eigenvalues of the baseline Gaussian covariance matrices in order to group them in the predefined number of groups, prior to the execution of the VQ clustering on each group separately. The largest eigenvalues are the most important for mixture grouping and their relative importance decreases with their value. For the aggregation of the value on the base on which the particular Gaussian component is to be grouped, we have proposed the usage of Ordered Weighted Average OWA aggregation operators (Janev et al., 2008). The idea is to give more weight to more significant (larger) eigenvalues in the aggregation process, thus optimising the OWA weights. They are to be applied to the particular eigenvalues vector $\lambda = (\lambda_1,\ldots, \lambda_p)$ in the following way:

$$OWA_{\omega}(\lambda_1,\ldots,\lambda_p) = \sum_{j=1}^{p} \omega_j \lambda_{\sigma(j)} \tag{11}$$

where $0 \le \lambda_{\sigma(1)} \le \ldots \le \lambda_{\sigma(p)}$. Depending on the OWA values, mixtures are divided into groups. The coefficients $\omega \in R^p$ satisfy the constrains that $0 \le \omega_j \le 1$ and they sum to one.

The OWA operators provide a parameterised family of aggregation operators which include many of the well known operators such as the maximum, the minimum, *k*-order statistics, median and the arithmetic mean. They can be seen as a parameterised way to interpolate between the minimum and the maximum value in an aggregation process. In this particular application, the applied operator should be somewhat closer to max(·) in order to favour more significant eigenvalues in the grouping process. The method to optimally obtain OWA coefficients introduced in (Yager, 1988) and used in (O'Hagan, 1988) is applied. The maxness $M(\omega) = \alpha \in [0,1]$ of the OWA operator is defined as:

$$M(\omega) = \sum_{j=1}^{p} \omega_j \frac{j-1}{p-1} \tag{12}$$

The idea is to maximise dispersion of weights $D(\omega)$ defined (O'Hagan, 1988) as

$$D(\omega) = -\sum_{j=1}^{p} \omega_j \ln(\omega_j) \tag{13}$$

thus obtaining the Constrained Nonlinear Programming (CNP) problem (O'Hagan, 1988). For finding the optimal weights ω_{opt}, any standard method can be used (Biggs, 1975), (Coleman et al., 1996). In the sequel, we give the baseline VQGS and EDGS algorithms as follows:

VQGS

Initialisation:

- For predefined n_{avr} and the overall number of mixtures M, calculate the number of clusters as: $N_{hpc} = |X| = \{M/n_{avr}\}$.
- Pick up at random (uniform distribution) N_{hpc} different centroids c_f f $\in \{1,\ldots,N_{hpc}\}$ from the set of overall M mixture centroids used. Assign to every centroid the identity covariance matrix $\Sigma_f = I$. Let Gaussian densities $X^{(0)} = \{\chi_f(c_f, \Sigma_f): f = 1,\ldots,N_{hpc}\}$ be initial hyper-mixtures.

Clustering:

Do the following, for predefined $\varepsilon > 0$

- To all mixtures θ_j, $j = 1,\ldots M$ assign a corresponding hyper-mixture $\chi^{(j)}$ in the current k-th iteration as: $\chi^{(j)} = \text{argmin } d(\theta_j, \chi)$, where $d(\cdot,\cdot)$ is symmetric KL divergence.
- Evaluate hyper-mixture parameters c_f and Σ_f using ML estimates (8), (9) and (10), to obtain $X^{(k)}$
- If any cluster "runs out" of mixtures, set $N_{hpc} = N_{hpc}$ - C for the next iteration, where C is the number of such clusters.

Until $D_{average} < \varepsilon$, for $D_{average}$ defined by (7).

EDGS:

Initialisation:

- Specify the number of groups G.
- Using any CNP method, obtain optimal OWE weights for predefined maxness $a \in [0,1]$ as: $\omega_{opt} = \text{argmax } D(\omega)$, satisfying constraints $M(\omega) = a$, that $0 \le \omega_j \le 1$ and they sum to one.
- For ω_{opt}, determine the group threshold vector (elements are group borders) $\tau = [\tau_{\max}{}^{(1)},\ldots \tau_{\max}{}^{(G-1)}]$, and set $\tau_{\min}{}^{(g+1)} = 0$, $\tau_{\max}{}^{(g)} = = \infty$. The group borders should satisfy the constraint: $\tau_{\max}{}^{(g+1)} = \tau_{\max}{}^{(g)}$, for $g = 1,\ldots,G\text{-}2$, where $\tau_{\max}{}^{(1)}$ is obtained heuristically.

Mixture Grouping:

For every $i = 1,\ldots,M$, for mixture θ_i do:

- Obtain eigenvalues $\lambda^{(i)} = (\lambda_1{}^{(i)},\ldots, \lambda_p{}^{(i)})$.
- Assign θ_i to the group giff: $OWE\omega_{opt}(\lambda^{(i)}) \in [\tau_{\min}{}^{(g)}, \tau_{\max}{}^{(g)})$

Perform baseline VQGS method on every group separately to obtain clusters with mixtures and corresponding hyper-mixtures.

The decoding process is given as follows

Decoding:

For all observations x_t, $t = 1,\ldots,N$, where N is the number of observations in the testing process do for every cluster C_k, $k=1,\ldots,N_{hpc}$ do:

- Evaluate log-likelihood $\ln f(x_t, \chi^{(k)})$, where $\chi^{(k)}$ is the hyper-mixture that corresponds to cluster C_k.
- If $\ln f(x_t, \chi^{(k)}) > \theta$, where θ is a predefined likelihood threshold, evaluate the exact likelihood for all mixtures that belong to the cluster C_k. Else, set all belonging mixture log-likelihoods to $\ln f(x_t, \Theta^{(k)})$ where $\Theta^{(k)})$ is the Gaussian mixture with centroid c_k and covariance matrix W_k defined by (10).

5. Conclusion

Both ASR and TTS systems described in this chapter have been originally developed for the Serbian language. However, linguistic similarities among South Slavic languages have allowed the adaptation of this system to other South Slavic languages, with various degrees of intervention needed.

As for ASR, adaptation to Bosnian and Croatian was very simple (due to extreme similarity of phonetics), whereas for Macedonian it was necessary to develop separate speech databases. The actual procedures used for ASR were almost identical in all cases. While well known algorithms were used for model training and testing, in this chapter only the original algorithms are presented. The VTN procedure based on the use of the iterative method and only static features for VTN coefficient estimation shows significant improvement in comparison to the common VTN procedure. The eigenvalue driven Gaussian selection significantly reduce computational load with minor increase of WER. Neither of the proposed algorithms is language dependent.

As for TTS, conversion of an arbitrary text into intelligible and natural-sounding speech has proven to be a highly language-dependent task, and the degree of intervention was variable and depended on specific properties of a particular language. For example, the simplicity of accentuation in Macedonian has allowed POS tagging and syntactic parsing to be avoided altogether, at the price of certain impairment in quality of synthesis. On the other hand, for Croatian and Bosnian, it was also necessary to build new accentuation dictionaries and to revise the expert system for POS tagging in order to assign words their appropriate accentuation, necessary for production of natural sounding speech.

It can be concluded that, in spite of the apparent language dependence of both principal speech technologies, some of their segments can be developed in parallel or re-used. The ASR and TTS systems described here are widely applied across the Western Balkans. In fact, practically all applications of speech technologies in the countries of the Western Balkans (Pekar et al., 2010) are based on ASR and TTS components described in this chapter.

5.1 Directions for future work

The team at the University of Novi Sad is a core of a greater multidisciplinary team in Serbia, whose aim is to further increase the quality of synthesised speech and the accuracy and robustness of ASR. The ultimate goal is to incorporate ASR and TTS into (multimodal) spoken dialogue systems, to expand ASR to larger vocabularies and spontaneous speech, not only in Serbian but in other South Slavic languages as well. Development of speech technologies for a language represents a contribution to the preservation of the language, overcoming language barriers and exploiting all the benefits coming from the use of speech technologies in one's native language.

6. References

Banerjee, A.; Merugu, S.; Dhillon, I. & Ghosh, J. (2005). Clustering with Bergman divergence, *Journal of Machine Learning Research*, Vol 6, pp. 1705-1749

Beutnagel, M.; Mohri, M. & Riley, M. (1999). Rapid unit selection from a large speech corpus for concatenative speech synthesis, *Proceedings of 6th EUROSPEECH*, pp. 607-610, ISSN 1018-4074, Budapest, Hungary

Benzeghiba, M.; De Mori R.; Deroo, O.; Dupont, S.; Jouvet, D.; Fissore, L.; Laface, P.; Mertins, A.; Ris, C.; Rose, R.; Tyagi, V. & Wellekens, C. (2006). Impact of Variabilities on Speech Recoginition, *Proceedings of 11th SPECOM* (*Speech and Computer*), St. Petersburg, Russia

Biggs, M. (1975). Constrained minimization using recursive quadratic programming. *Dixon LCW, Szergo GP (Eds.) Towards global optimization*. North-Holland, Amsterdam, pp. 341–349

Bocchieri, E. (1993). Vector quantization for efficient computation of continuous density likelihoods. *Proceedings of ICASSP*, Minneapolis,MN, Vol 2, pp. II-692–II-695

Brill, E. (1992). A simple rule-based part of speech tagger. *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pp. 152-155, Trento, Italy

Coleman, T. & Li, Y. (1996). An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM J Optim* 6, pp. 418–445

Delić, V.; Pekar, D.; Obradović, R.; Jakovljević, N. & Mišković, D. (2007). A Review of AlfaNum Continuous Automatic Speech Recognition System, *Proceedings of 12th SPECOM (Speech and Computer)*, pp. 702-707, ISBN 6-7452-0110-x, Moscow, Russia, October 2007

Delić, V. (2000). Speech corpora in Serbian recorded as a part of AlfaNum project, *Proceedings of 3th DOGS (Digital Speech and Image Processing),* pp. 29-32, Novi Sad, Serbia, October 2000, Novi Sad

Dutoit, T. (1997). *An Introduction to Text-to-Speech Synthesis*, Kluwer Academic Publishers, ISBN: 0-7923-4498-7, Dordrecht/Boston/London

Hajič, J. & Hladká, B. (1998). Czech language processing – POS tagging. *Proceedings of 1st International Conference on Language Resources and Evaluation*, pp. 931-936, Granada, Spain

Gillick, S. & Cox, S. (1989). Some Statistical Issues in the Comparison of Speech Recognition Algorithms, *Proceedings ICASSP,* pp. 532-535

Goldberg, J. & Roweis, S. (2005). Hierarchical clustering of a mixture model, *Proceedings of NIPS 2005*, December 5, Vancouver

Gouvea, E. & Stern, R. (1997). Speaker Normalisation through Formant Based Warping of Frequency Scale, *Proceedings of EUROSPEECH*, pp. 1139-1142, Rhodes, Greece

Jakovljević, N.; Mišković, D.; Sečujski, M. & Pekar, D. (2006). Vocal Tract Normalisation Based on Formant Positions, *Proceedings of IS-LTC*, Ljubljana, Slovenia

Jakovljević, N.; Sečujski, M. & Delić, V. (2009). Vocal Tract Length Normalisation Strategy Based On Maximum Likelihood Criterion, *Proceedings of EUROCON*, pp. 417-420, ISBN 978-1-4244-3861-7, St. Peterburg, Russia

Jakovljević, N. (2009). *Improvement of ASR performance using Vocal Tract Length Normalisation* (*M.Sc. thesis)*, Faculty of Technical Sciences, University of Novi Sad, Serbia (in Serbian)

Janev, M.; Pekar, D.; Jakovljević, N. & Delić, V. (2008). Eigenvalues driven gaussian selection in continuous speech recognition using HMMs with full covariance matrices. *Applied Intelligence*, *Springer Netherlands,* DOI: 10.1007/s10489-008-0152-9, (Print, accepted) December 2008, ISSN 0924-669X, 1573-7497 (Online, available) http://www.springerlink.com/content/964vx4055k424114/

Jurafsky, D. & Martin, H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice-Hall, ISBN-10: 0131873210, Upper Saddle River, NJ.

Knill, M.; Gales, F. & Young J. (1996). Use of Gaussian selection in large vocabulary continuous speech recognition using HMMs, *Proceedings of Int. Conf. Spoken Language Processing*

Knill, M.; Gales, F. & Young, J. (1999). State based Gaussian selection in large vocabulary continuous speech recognition using HMMs, Mar 1999, Vol 7, Issue 2, pp. 152-161

Lee, L. & Rose, R. (1996). Speaker Normalisation using Efficient Frequency Warping Procedures, *Proceedings of ICASSP*, pp. 353-356

Lindo, Y.; Buzo, A. & Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Trans Commun COMM* 28, pp. 84–95

Miguel, A.; Lleida, E.; Rose, R.; Buera, L.; Saz, O. & Ortega, A. (2008). Capturing Local Variability for Speaker Normalisation in Speech Recognition, *IEEE Transactions on Audio, Speech and Language Processing*, pp. 578-593

Molau, S. (2003) Normalisation of Acoustic Feature Space for Improved Speech Recognition, (PhD Thesis), RWTH Aachen, Germany

Obradović, R. & Pekar, D. (2000). C++ Library for Signal Processing. *Proceedings of DOGS (Digital Speech and Image Processing)*, Novi Sad, Serbia, pp. 67-70.

O'Hagan, M. (1988). Aggregating template or rule antecedents in real time expert systems with fuzzy set logic. *Proceedings* of the 22-th annual IEEE Asilomar conferences on signals, systems and computers, Pacific Grove, pp. 681–689

Oravecz, C. & Dienes, P. (2002). Efficient stochastic part-of-speech tagging for Hungarian. *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pp. 710-717, Las Palmas, Spain

Riley, M. D. (1989). Some applications of tree-based modeling to speech and language indexing. *Proceedings of the DARPA Speech and Natural Language Workshop*, pp. 339-352. Morgan Kaufmann

Sečujski, M. (2002). Accentuation dictionary of Serbian language intended for text-to-speech synthesis (in Serbian), *Proceedings of 4th DOGS (Digital Speech and Image Processing)*, pp. 17-20, Bečej, Serbia, May 2002, Publisher: FTN Novi Sad

Sečujski, M.; Obradović, R.; Pekar, D.; Jovanov, Lj. & Delić, V. (2002). AlfaNum System for Speech Synthesis in Serbian Language. *Proceedings of TSD (Text, Speech and Dialogue)*, pp. 237-244, ISBN 3-540-44129-8, Brno, Czech Republic, September 2002. *Lecture Notes in Artificial Intelligence*, *Springer-Verlag Berlin Heidelberg,* LNAI 2448, pp. 237-244, ISSN 0302-9743

Sečujski, M. (2005). Obtaining Prosodic Information from Text in Serbian Language, *Proceedings of EUROCON*, pp. 1654-1657, ISBN 86-7466-218-8 (AM), Belgrade, Serbia, November 2005

Sečujski, M.; Delić, V.; Pekar, D.; Obradović, R. & Knežević, D. (2007). An Overview of the AlfaNum Text-to-Speech Synthesis System, *Proceedings of 12th SPECOM (Speech and Computer)*, pp. Ad.Vol. 3-7, ISBN 6-7452-0110-x, Moscow, Russia, October 2007

Sečujski, M. (2009). *Automatic Part-of-Speech Tagging of Texts in Serbian Language* (*PhD thesis)*, Faculty of technical Sciences, University of Novi Sad, Serbia

Sepesy Maučec, M.; Rotovnik, T. & Zemljak, M. (2003). Modelling Highly Inflected Slovenian Language. *International Journal of Speech Technology, Springer, the Netherlands,* Vol. 6, No. 3, pp. 245-257, ISSN 1381-2416

Shinoda, K. & Lee, C. (2001). A structural Bayes approach to speaker adaptation. *IEEE Trans Speech Audio Process* 9(3), pp. 276–287

Simonin, J.; Delphin, L. & Damnati, G. (1998). Gaussian density tree structure in a multi-Gaussian HMM based speech recognition system. *Proceedings of 5th Int. Conf on Spoken Language Processing*, Sydney, Australia

Tokuda, K.; Yoshimura, T.; Masuko, T.; Kobayashi, T. & Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis, *Proc. of ICASSP*, pp. 1315-1318, Istanbul, Turkey

Uebel, L. & Woodland, P. (1999). An Investigation into Vocal Tract Length Normalisation, *Proceedings of EUROSPEECH,* pp. 2527-2530

Webb, A. (1999). *Statistical Pattern Recognition*, Oxford University Press Inc, ISBN 0-340-74164-3, New York, USA

Welling, L.; Kanthak, S. & Ney, H. (1999). Improved Methods for Vocal Tract Normalisation, *Proceedings of ICASSP,* pp. 761–764, Phoenix, USA

Woszczyna, M. & Fritsch, J. (1997). Codebuch übergreifende bucket-boxintersection zur schnellen Berechnung von Emissionswahrscheinlichkeiten im Karlsruher VM-Erkenner. Verbmobil

Yager, R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Trans Syst Man Cybern* 18, pp. 183–190

Young, S.; Odell, J. & Woodland, P. (1994). Tree-based state tying for high accuracy acoustic modelling, *Proceedings of the workshop on Human Language Technology*, pp. 307-312, Association for Computational Linguistics, ISBN:1-55860-357-3, Plainsboro, NJ.