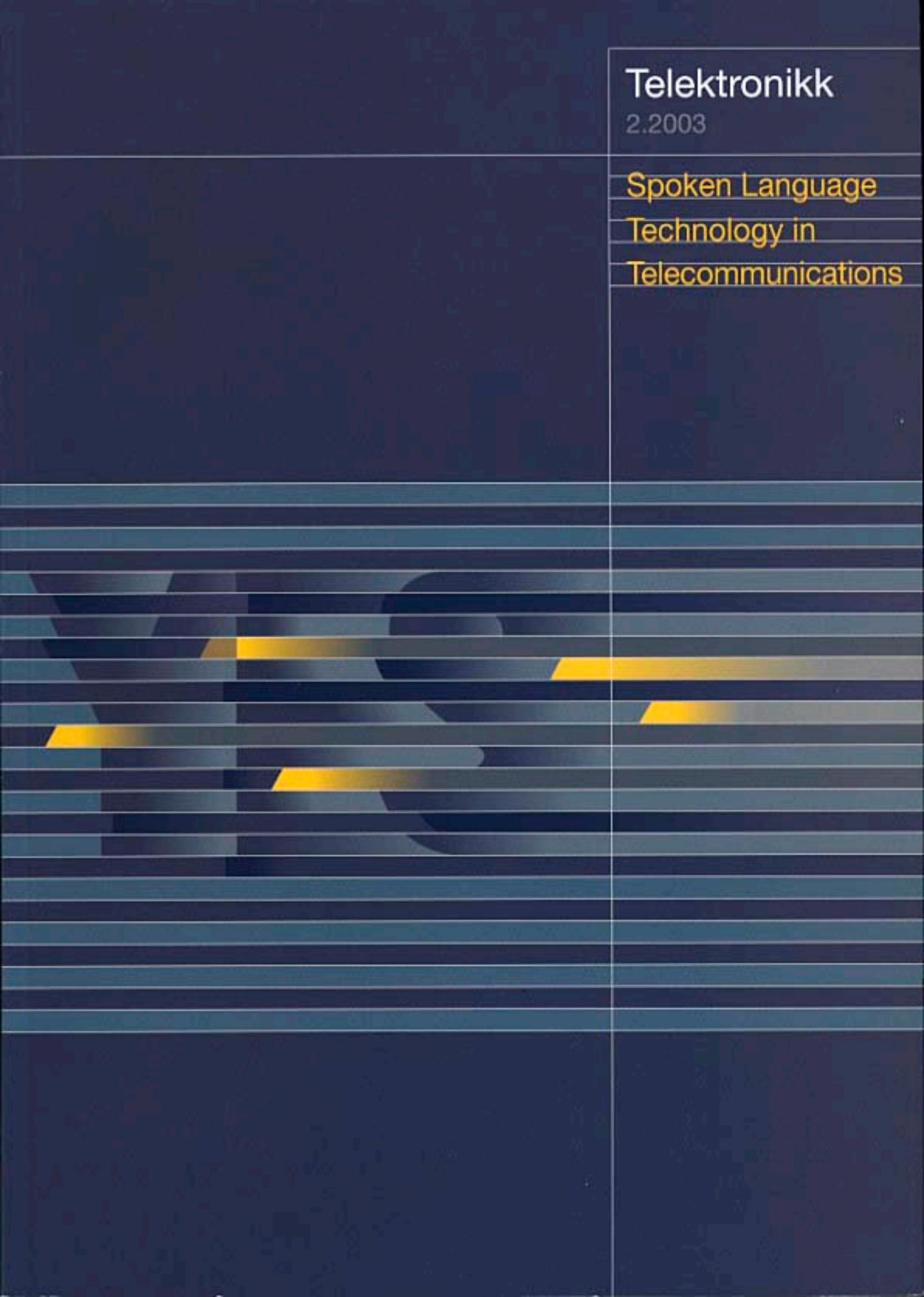


Telektronikk

2.2003

Spoken Language
Technology in
Telecommunications

The background of the page is a dark blue gradient. A series of horizontal lines in varying shades of blue and grey run across the page. Several bright yellow horizontal bars are interspersed among these lines, creating a rhythmic, abstract pattern. The text is positioned in the upper right corner, with a white border around it.

Contents

Telelektronikk

Volume 99 No. 2 – 2003

ISSN 0085-7130

Editor:

Ola Espvik

Tel: (+47) 913 14 507

email: ola.espvik@telenor.com

Status section editor:

Per Hjalmar Lehne

Tel: (+47) 916 94 909

email: per-hjalmar.lehne@telenor.com

Editorial assistant:

Gunhild Luke

Tel: (+47) 415 14 125

email: gunhild.luke@telenor.com

Editorial office:

Telenor Communication AS

Telenor R&D

NO-1331 Fornebu

Norway

Tel: (+47) 810 77 000

telelektronikk@telenor.com

www.telenor.com/rd/telelektronikk

Editorial board:

Berit Svendsen, CTO Telenor

Ole P. Håkonsen, Professor

Oddvar Hesjedal, Director

Bjørn Løken, Director

Graphic design:

Design Consult AS (Odd Andersen), Oslo

Layout and illustrations:

Gunhild Luke and Åse Aardal,

Telenor R&D

Prepress and printing:

Optimal as, Oslo

Circulation:

3,200

Spoken Language Technology in Telecommunications

1 Guest Editorial; *Narada Warakagoda and Ingunn Amdal*

3 Introduction; *Narada Warakagoda and Ingunn Amdal*

Overview Articles

6 Speech Technology: Past, Present and Future; *Torbjørn Svendsen*

19 Computational Linguistics Methods for Machine Translation in Telecommunication Applications?; *Torbjørn Nordgård*

23 Speech Recognition – a Tutorial and a Commentary; *Frank K Soong and Biing-Hwang Juang*

30 An Overview of Text-to-Speech Synthesis; *Per Olav Heggveit*

Automatic Speech Recognition

45 Auditory Based Methods for Robust Speech Feature Extraction; *Bojana Gajic*

59 Adaptation Techniques in Automatic Speech Recognition; *Tor André Myrvoll*

70 Pronunciation Variation Modeling in Automatic Speech Recognition; *Ingunn Amdal and Eric Fosler-Lussier*

Text-to-Speech Synthesis

83 The ‘Melody’ in Synthetic Speech: What Kind of Phonetic Knowledge Can Be Used to Improve It?; *Jørn Almberg*

94 Prosodic Unit Selection for Text-to-Speech Synthesis; *Jon Emil Natvig and Per Olav Heggveit*

Multimodal User Interfaces

104 Speech Centric Multimodal Interfaces for Mobile Communication Systems; *Knut Kvale, Narada Warakagoda and Jan Eikeset Knudsen*

119 Multimodal Interaction – Will Users Tap and Speak Simultaneously; *John Rugelbak and Kari Hammes*

Real World Speech Technology Based Applications

126 A Norwegian Spoken Dialogue System for Bus Travel Information – Alternative Dialogue Structures and Evaluation of a System Driven Version; *Magne Hallstein Johnsen, Tore Amble and Erik Harborg*

132 *Terms and Acronyms*

Guest Editorial

NARADA WARAKAGODA AND INGUNN AMDAL



Narada Warakagoda



Ingunn Amdal

Front cover: **Spoken Language Technology in Telecommunications**

A variety of persons, a variety of ways of talking – and often a variety of languages. A machine that responds to a spoken language must have the ability to adopt to all such diversities within the repertoire of instructions/communications that can be understood. The basic issue is yes or no to accepting an instruction upon which a desired action takes place.

A machine may sense your presence in a certain setting and take the responsibility to guide you to a number of predefined “destinations” – on the way to which a decision-based communication may take place. A person can choose not to react to the machine speaking, but the machine must react if it concludes “yes – understood” to spoken instructions delivered all through the various segments of a service/services. The artist Odd Andersen visualises this basic process by a dynamic Yes as we move through the spectrum of possibilities on our way to fulfilling our service needs.

Ola Espvik, Editor in Chief

Imagine a world without speech and language! That cannot be the world you and I know. Imagine a world of telecommunications without speech and language! That cannot be a world meant for humans.

Humans invented speech and language to communicate with each other. They invented telecommunications to use speech and language over larger distances. In fact, the first practical telecommunications system of the world, telegraphy, did not support transmission of speech. It could only transmit text messages using the *Morse code*. Invention of telephony in the beginning of the last century put speech in the centre of telecommunications, something which led to widespread acceptance and use of telecommunications systems throughout the world. The speech centric nature of telecommunications prevailed for many years, until the late 1990s, when *digital data* began to become the focus of the industry.

Undoubtedly, the paradigm shift from speech centrality to data centrality, also known as convergence of speech and data communication systems, represents a positive development and has created a lot of new possibilities. However, the most significant consequence of this paradigm shift is the introduction of machines that possess some kind of intelligence to the telecommunication grid. This in turn has resulted in at least two new basic scenarios in bipartisan telecommunications, namely, human-to-machine and machine-to-machine communications. These two new scenarios together with the old, but still important scenario of human-to-human communication, make the basis of all imaginable possibilities in modern telecommunications.

Even though the emergence of data centric communication seems to have pushed the speech technology to a corner in the telecommunications arena, this is actually not the case at all. We should not forget that speech based human-to-human communication is still a major part of telecommunications. This is not going to change significantly as long as speech remains to be humans' preferred mode of communication with each other. Nevertheless, technologies such as speech coding, which provides the base for speech based telecommunications, have reached a saturation point and therefore it is difficult to expect a significant growth in this area, at least

in a technological sense. In the meantime, however, new scenarios in telecommunication, human-to-machine and machine-to-machine communications, have made speech and language technologies even more important.

A typical example of human-to-machine communication would be the scenario where a user requests a service from a machine using a communication device. A fundamental problem in such a situation is the fast and efficient understanding of each other's goals, thoughts and intentions. Arguably speech is the key to solving this problem since it is the most developed and efficient communication mechanism possessed by humans. Using speech for better human-machine interaction has been a dream for a long time. However, the idea was limited to science fiction until the 1970s. But now, the key technologies that realize this dream, *speech recognition* and *speech synthesis*, are mature technologies used in widespread commercial applications. Speech recognition makes it possible for the machine to know the words the user utters to express his thoughts and intentions, whereas speech synthesis converts the information the machine gathers, for example by searching a database, back to speech.

Even though speech has its strengths in many situations, there are occasions where speech is not the most suitable candidate medium for providing/acquiring information to/from a machine. However, if speech is combined with other types of input or output channels, this disadvantage can be avoided. Multimodal processing makes such combinations possible and hence allows a high quality human-machine interaction.

Human-machine interaction can be seen as cooperation between a human and a machine to perform a certain task. Unless the task is very simple, this cooperation needs to extend over several steps and in each of these steps either the user or the machine queries the respective counterpart to get a response. In other words, there is a dialogue between the user and the machine. During the dialogue, the machine should at least be able to understand the user's queries and formulate answers, to generate queries for the user, to contact the back-end applications such as databases and to decide what to do next. Dialogue control is a technology that allows us to endow the machine with such an ability.

While speech recognition, speech synthesis, multimodal processing and dialogue control are essential for efficient and natural human-machine communication, another set of technologies related more to the written language are of paramount importance to the future machine-to-machine communication.

In the early era of machine-to-machine communication, the focus was on transferring a chunk of bytes from one machine to another without considering the meaning hidden behind these data. However, this attitude is changing with the emergence of new open communication paradigms such as XML based web services. In one modern view, the purpose of machine-to-machine communication would be to make use of data (and other resources) in one machine to provide the information required by another machine. Usually, the information sought is related to the meanings of the available data and therefore a processing at semantic level is needed to serve such requests. If the data were available in a structured form such as tables in relational databases, then this would be relatively easy, at least for restrictive tasks. But unfortunately more and more of machines exposed to the Internet today are serving unstructured natural language (NL) documents or “weakly” structured HTML or XML documents. The emerging semantic web paradigm is an effort that tries to change this trend by marking-up semantics of the web page contents. However, it is way too unrealistic to assume that at least a substantial part of the existing or future NL documents on the Internet will be (manually) structured by their authors to a sufficiently fine level for their semantics to be readily utilized by other machines. The only way out of this problem is to enrich the communication chain by introducing components that perform online or offline automatic semantics processing. This is exactly where the language technologies play their role.

Natural language understanding is a technology which enables a machine to read through an NL document and extract its meaning, so that advanced queries can be answered. However this is not a very mature technology and the substitutes

are less powerful information retrieval or question answering techniques which are based on shallow, often statistical analysis of texts. If several languages are involved in communication, then automatic translation technology is needed to match the request, data and response with each other.

Spoken language technology is a vast area which spans from speech recognition and speech synthesis through multimodal processing to dialogue processing, and maintains close ties with natural language understanding, information retrieval and automatic translation. This issue of *Teletronikk* contains a collection of articles which disclose the internal details of these core technologies as well as how they can be utilized to create innovative telecommunications applications.

The success of telecom applications and services offered to the users depends heavily on whether they make a real difference in the users’ day-to-day lives. Further, everybody in society, irrespective of their technical capabilities, should be able to get the benefit of today’s hugely invested telecommunications infrastructure. The key to achieving this is the introduction of more intelligence to the network – where appropriate, whether it is between the user and the system or between the system units. Unfortunately, this does not come cheaply; hard work based on a deep understanding of the state-of-the-art is required to achieve the necessary technology breakthroughs. Contrary to the widely held (mis)conception, we believe that this is a common responsibility of all actors in the telecommunications industry, including operators, equipment manufacturers and research organizations.

We hope that this issue of *Teletronikk* will give the reader an insight into how spoken language technology can be used to make telecommunications services more intelligent and hence more accommodating.



Introduction

NARADA WARAKAGODA AND INGUNN AMDAL



Narada Warakagoda (38) is a research scientist at Telenor R&D. He graduated from the Norwegian University of Science and Technology as Siv.Ing. (MSc) in 1994 and Dr.Ing. (PhD) in 2001. During his work towards the PhD degree, he developed automatic speech recognition algorithms based on the theory of nonlinear dynamical systems. After joining Telenor R&D, his research began to orientate more towards speech technology based telecom applications. Recently he has been involved in several projects on speech centric multimodal systems. His current research interests are multimodal interfaces which can exhibit human-like intelligent and emotional behaviour.

*narada-dilp.warakagoda
@telenor.com*



Ingunn Amdal (38) graduated from the Norwegian University of Science and Technology (NTNU) with a Siv.Ing. (MSc) in 1989 and a Dr.Ing. (PhD) in 2002. In 1990 she started as research scientist at Telenor R&D working on loudspeaking telephones and acoustic echo-cancellation. She joined the newly formed speech technology group at Telenor R&D in 1994 working with automatic speech recognition, dialogue design, and database collection. The topic of her PhD thesis was pronunciation modelling and her current interests include design, representation and evaluation of spoken and multimodal dialogue systems, user tests and spontaneous speech processing.

ingunn.amdal@telenor.com

1 Introduction

Spoken language played a crucial role for the survival of humans in the early stages of their history. Technological development in spoken language for inter-human (tele)communication was critical for the impressive advancement of human civilization in recent history. Spoken language technology will play an even more important role in our battle to improve living standards in the modern day and the future information society.

The concept of Information Society (IS) has attracted huge interest from every corner of the world. In particular, the European Union (EU) has taken it into consideration in policy making and developed a clear vision of such a society in a 5–10 year perspective. Further, the Information Society Technologies Advisory Group (ISTAG) of the EU has identified “Ambient Intelligence” as the main feature of the IS in which technology will help people in their everyday life, both for business and pleasure. In this vision personal agents interact with their owner as well as other agents and humans. When the owner is busy, the agent will communicate on behalf of the owner and settle small affairs, but let important messages through. In the ISTAG scenarios the spoken language capability of the agents is used extensively, but it has been taken for granted!

However, realization of these scenarios requires significant enhancements of the spoken language and related technologies, even though these technologies are mature enough for certain types of applications even today. These enhancements include increased accuracy of speech recognition under real world conditions, improved quality of speech synthesis, improved quality and robustness of dialogue handling, increased utilization of multimodality and betterment of the quality of machine translation etc. In order to give an idea of these areas together with the possibilities of the spoken language technology both today and in five years’ time, we will present two scenarios. The technology building blocks and associated research issues of these scenarios are covered in detail by the subsequent articles in this collection. The connection between each of the issues of interest and the corresponding articles that go into details of the issue concerned are indicated as references in the scenario descriptions. Therefore, the scenarios presented in the following two sections can be considered as an overlay that puts the articles in this collection into perspective.

2 A Day With Speech/ Language Technology in 2003

Rune is a 35 year old lawyer who works for a big oil company. Next week he will have to attend a business meeting with one of their collaborating companies in Russia. He therefore wants to find some information about an area close to Moscow where he plans to stay. He has used a web search engine and found several pages which contain relevant information about this area. Unfortunately, these pages are in Russian. Thanks to automatic translation services offered by certain websites (e.g. Babelfish) he solves the problem within seconds [2]. Even though he gets most of the information from the Russian-to-English translated pages, the quality is not very good. Therefore he wants to talk to one of his colleagues who has experiences in doing business in Russia. Rune vaguely remembers her last name, but does not know her telephone number. Unfortunately, it is almost noon and most of the secretaries at the switchboard are out for lunch. But, in this case too, he just calls the automatic switchboard service of the company which is operational 24 hours a day, seven days a week [1] [12]. The service recognizes the name Rune utters already on the second try and asks whether the call should be forwarded automatically to that person using a synthetic voice. The synthesized voice is understandable though not quite natural for this infrequent name [4]. He says yes and is automatically forwarded to the person he wishes to talk to.

After receiving the relevant information, he has to take a short leave due to an appointment with his radiologist. Rune has a suspected broken rib from an accident on a previous skiing trip. The radiologist finishes the examination very quickly as he uses an automatic dictation system to take the notes, something which allows both of his hands and eyes to be free [3]. After the hospital visit Rune has to attend an important meeting lasting the whole afternoon. On his way back home Rune wants to plan his trip to Moscow. Even though the current time is 4.30 in the afternoon it is not a big problem. He just calls the automatic voice operated flight and train information services. These services have no negotiation or understanding capability, but the trip is planned and tickets are booked when he arrives home [12]. Rune was a bit puzzled, however, because in each of the above services, the synthetic voice answering for the different companies was the same [4].

3 Norwegian Tourist in Beijing for the Olympics in 2008

John is on his way to see the Olympics in Beijing. His fellow travellers are already there; he had to join a couple of days later because he had an important contract to finish. Now everything is settled and his mind is totally off work and ready to enjoy the sporting event, the city of Beijing, which he has never visited before, and the company of his friends. He takes with him his “multi-everything terminal”, a PDA-type device which he decided to buy recently. One reason for buying it was exactly this trip. Thanks to the open standards, this device is fully compatible with the devices his friends have got. John had no time to try it out, but his wife uploaded Beijing software before he left.

On the long haul flight to Beijing, John has nothing else to do than to configure and try it out. His family and friends will expect him to use the device to keep them updated on the Olympic events and all the exciting things he will encounter. The first step in configuring the device is to use speaker verification “It’s me” to load a personal profile with advanced adaptation features [6] for the best speech recognition performance. He regularly uses speech dictation software at work to cure his mouse strain injury, but forgot to upload his personal versions of acoustic models, pronunciation and speaking style. Anyway, this device has new online adaptation that is improved while he is speaking and is saved with his personal profile. There is no need for specific adaptation training like in the older systems. The noise in the cabin is annoying, but noise robust features [5] make the speech recogniser work properly.

First of all he tries out the tourist guide application which is equipped with advanced speech technology. Even if he has not used this guide before it is easy and intuitive to communicate through the friendly avatar structuring the information needed into a personalized style guide [10]. The terminal is multimodal to show maps and tables of information while still maintaining the rich capabilities of spoken language interaction. The multimodal terminal manufacturer did make it through the last crack in the telecommunication industry since they decided to make their user interface design stand out from the rest of the competitors [11].

John arrived at the Beijing airport next morning. At the taxi information at the airport a low cost speech technology system is available where the tourists can ask where to go without speaking Mandarin. Several languages are available, but not Norwegian – John chooses English. Lexical adaptation to non-natives [7] is available and for the medium vocabulary of streets, points-of-

interest and hotels in Beijing this is sufficient to get a reasonable performance. The translated request is printed out in both languages, one for the traveller and one for the driver. On the way into the city John uses the GPS location to follow the route on his map. Points-of-interest are also highlighted as their drive proceeds. The avatar asks if he wants a guided tour as they drive, and a pleasant text-to-speech synthesis voice [8] reads selected information about the things that they pass by. He has several options for personal favourites in voices for TTS. For reading longer texts about historical places “emotional” speech is necessary.

After having settled into the hotel John starts to plan his day. He is really a well equipped tourist confident on managing his tourist affairs helped by his PDA-based tourist guide. He has plans to join his buddies today, but before that he wants to walk around the city with his “electronic assistant” to get the feel of Beijing. Again the noise robust features are essential for the performance of the device in the heavy traffic noise, and the babble of public places [5]. The machine translation capabilities of the device makes it possible to talk to people the tourist meets, where the avatar works as a translator. The machine translation properties [2] are working well enough to order food at restaurants and John has a nice meal in a local dumpling restaurant. The device has a tone recognition module for the best performance on Mandarin [3].

At a cafe our tourist decides to listen to a voice message that arrived when he was busy ordering food. It is from his brother on trip on the west coast of Norway. At first he is puzzled to hear that this certainly sounds like his brother’s voice but with a west coast intonation. John calls him up and then his brother enthusiastically tells about a west coast friend who has got his hands on TTS software with intonation patterns from different dialects [9].

Now it is time for John to meet his friends and he locates them on the map, the claimed compatibility of the device works. He sends a voice message to all of them to say that he has arrived safe and sound and is ready to meet them. The avatar locates a nice pub close to the football stadium where two of John’s friends have just finished watching Norway losing to France in their first match. The friends get the message converted to text because they would not have a chance to hear anything in the noise of the French singing [3] [10].

It is quite a distance to the stadium and our traveller decides to take a taxi – they are quite cheap even during the Olympics. He lets the avatar speak to the taxi-driver. The taxi driver answers

(through the avatar) that John is his third passenger with this kind of device and that he is getting quite used to them [10]. What our two men in the taxi do not know is that there is a traffic jam ahead because of an accident on the planned route. The traffic surveillance system is sending out messages, fortunately in a format compatible with the device. The avatar notifies the driver and they discuss how to proceed. John gets a bit annoyed that he is left out of the discussion. He breaks into the conversation and the avatar informs him that they were discussing where to buy the cheapest petrol.

In front of the stadium it is impossible to see John's mates in the crowd. "How can I find them", John cries out in despair, but the device notifies his two friends and guides them to each other. In the pub after the usual greetings, they order beer with the help of the translation module in the PDA [2] [3]. It automatically switches to a speech synthesis voice similar to the waiter and the avatar acts as a mediator to carry on the conversation with the waiter. One of his friends who has another dialect wants to try out the translation. But the pronunciation module [7] detects this and merges the appropriate pronunciation rules with the stored pronunciations. After getting the beer and updating the plans for tomorrow, they find that the night is still young and they do not want to return to the hotel yet. "What's on tonight", John asks the avatar, which quickly locates the specific Olympic module that is free to download and shows events nearby. There are many things happening and the small screen cannot show them all. "You have to say what kind of event you would like or where", the avatar says. "Something with a local flavour, but close to here", one of them asks and the result is a list of Chinese cultural events in the neighbourhood [10]. They decide to go to a mini-version of a Beijing Opera, something they have never seen before. "But don't store this choice in my profile before I know I like it", is the last message to the avatar before they turn it off for the night.

At the entrance they are all delightfully surprised to find the fourth friend waiting for them together with an old acquaintance of them all. "I ran into her on the Tienamen square, what a coincidence! As your avatar was in "group" mode it notified us where you were going, so we decided to join!"

4 References

- 1 Svendsen, T. Speech Technology: Past, present, and future. *Teletronikk*, 99 (2), 6–18, 2003. (This issue)
- 2 Nordgård, T. Computational linguistics methods for machine translation in telecommunications applications? *Teletronikk*, 99 (2), 19–22, 2003. (This issue)
- 3 Soong, F K, Juang, B-H. Speech Recognition – a tutorial and a commentary. *Teletronikk*, 99 (2), 23–29, 2003. (This issue)
- 4 Heggveit, P O. An overview of text-to-speech synthesis. *Teletronikk*, 99 (2), 30–44, 2003. (This issue)
- 5 Gajic, B. Auditory based methods for robust speech feature extraction. *Teletronikk*, 99 (2), 45–58, 2003. (This issue)
- 6 Myrvoll, T A. Adaptation techniques in automatic speech recognition. *Teletronikk*, 99 (2), 59–69, 2003. (This issue)
- 7 Amdal, I, Fosler-Lussier, E. Pronunciation variation modeling in automatic speech recognition. *Teletronikk*, 99 (2), 70–82, 2003. (This issue)
- 8 Almberg, J. The "melody" in synthetic speech: What kind of phonetic knowledge can be used to improve it? *Teletronikk*, 99 (2), 83–93, 2003. (This issue)
- 9 Natvig, J E, Heggveit, P O. Prosodic unit selection for text-to-speech synthesis. *Teletronikk*, 99 (2), 94–103, 2003. (This issue)
- 10 Kvale, K, Warakagoda, N, Knudsen, J E. Speech centric multimodal interfaces for mobile communication systems. *Teletronikk*, 99 (2), 104–118, 2003. (This issue)
- 11 Rugelbak, J, Hammes, K. Multimodal interaction – will users tap and speak simultaneously? *Teletronikk*, 99 (2), 119–125, 2003. (This issue)
- 12 Johnsen, M H, Amble, T, Harborg, E. A Norwegian spoken dialogue system for bus travel information. Alternative dialogue structures and evaluation of a system driven version. *Teletronikk*, 99 (2), 126–132, 2003. (This issue)

Speech Technology: Past, Present And Future

TORBJØRN SVENDSEN



Torbjørn Svendsen (48) received his *Siv. Ing. (MSc)* and *Dr. Ing. (PhD)* degrees in Electrical Engineering from the Norwegian University of Science and Technology (NTNU) in 1979 and 1985, respectively. The topic of the theses was speech coding. 1981–1988 he was a scientist at SINTEF DELAB working primarily on speech coding and recognition. In 1988 he joined the Dept. of Telecommunications at NTNU, where he currently is professor. He has been visiting professor at Griffith University and Queensland University of Technology, and has had several research visits to AT&T Bell Laboratories and AT&T Labs. Dr. Svendsen's main field of research is speech processing, encompassing speech recognition, speech coding and speech synthesis. He is a member of IEEE, the European Speech Communication Association (ESCA) and NORSIG.

torbjorn@tele.ntnu.no

Speech technology is intended to enable human-machine communication by voice. This includes making computers able to understand human speech, *speech recognition*, to produce intelligible and natural sounding speech, *speech synthesis*, and to determine who is speaking, *speaker recognition*. Unlike many other technology areas, speech technology is closely linked to an understanding of basic human properties. Without knowledge of the fundamentals of human speech production and perception, progress in speech technology will be difficult to achieve. In this paper we start out with an overview of human speech communication before describing the main trends in the development of speech technology, from the early efforts of mimicking human speech production up till today's, mainly data-driven statistical approaches. Although the progress in speech technology performance has been great over the last 25 years, the technology is far from perfect. Many useful products and services employing speech technology exist today, but significant research issues need to be resolved before computers can approach the performance of human listeners and talkers.

Introduction

The capacity for speech communication is a trait that distinguishes man from other species. Consequently, speech has also fascinated humans for centuries. In ancient times, objects that seemingly could produce and understand speech were attributed with having supernatural powers. One such phenomenon was the Greek oracles, which would give cryptic answers and predictions to the questions presented. Today, we know that the voice of the oracle emanated from a person hidden inside the oracle, speaking into a tube that led into the oracle's head.

In more recent times, speech enabled objects are still desired, although the supernatural notions are long abandoned. Speech is a simple means of communication for humans, and has the advantage that communication can be undertaken while your hands and eyes are busy with other tasks. Speaking and listening does not require that you learn motor skills like using a keyboard, and does not even require that you are able to read and write.

Speech technology can enable us to communicate with machinery by voice. This includes the technology that can enable machines to produce intelligible and natural sounding speech: *speech synthesis*, and the technology for recognizing or understanding what we speak: *speech recognition*. Combining these technologies is necessary in order to produce *spoken dialogue systems*, systems where the interaction between user and machine is based on speech.

In spite of intensive research efforts over nearly half a century, making truly speech enabled machines remains an elusive goal. Although the technology has come a long way since the first efforts at producing electronic machinery for this task, the current state-of-the-art systems can be

described as being at an early adolescent evolutionary stage. The simple tasks are well mastered. More complex tasks are also mastered when the circumstances are favorable and the environment well controlled, but the performance tends to fail when this is not the case. The technology that will enable spoken human-machine communication to be as effortless and reliable as spoken communication between humans, for all acoustic environments, topics of conversation and speaker accents is still far from reality. Indeed, until we have a deeper and more complete understanding of the human speech communication process, and of how technology can be applied to speech communication, speech technology will only be successful in restricted application domains.

What is it that makes speech communication such a difficult task for machines? Most humans master the process of speech communication without much drama, although learning to speak and to understand speech can take many years. Once learned, the human capacity for e.g. recognizing speech in noisy environments is quite amazing. Near perfect speech recognition in environments where the noise level exceeds the speech level is something most people do without too much effort. An automatic speech recognizer on the other hand will be rendered nearly useless if the SNR drops below 10–15 dB. Changes in speech patterns, topic shifts etc. are also handled without problems by humans while the machines have huge difficulties with situations that deviate from a well-defined and well-known setting.

In this paper, we will take a look at some of the main technology development trends in different speech technology areas. We will begin with a short description of the basics of speech communications, speech production and perception

before moving to the technologies that aim to enable machines to speak, and to recognize (and understand) speech.

Speech Communication

Let us take a look at the steps involved in speech communication. In Figure 1, a communication-theory inspired interpretation of the steps involved in the speech communication chain is depicted [16]. The initial step is generation in our mind of a notion or an idea that we wish to convey. This is represented as an output, M , from a message source, with an associated probability, $P(M)$. The message will need to be formulated in words, a transformation that is dependent on the semantics, syntax and grammar of the language. In the speech chain this is illustrated by transmitting the message through the linguistic channel, producing a word sequence W . Since a message can be formulated in many ways, the mapping is described by a probability, $P(W|M)$. The word sequence then triggers neuromuscular activity in the articulatory channel. This produces the spoken message, which is a sequence of sounds, S , radiating as an acoustic wave from the speaker's mouth. The articulatory channel will be different for every individual speaker; the shape and length of the vocal tract, the dimensions of the vocal folds and the muscles that control the articulators are different for each individual. This makes it impossible for two speakers to produce identical waveforms, indeed, it is near impossible for a speaker to reproduce a particular realization. The mapping of words to sounds is ruled by the probability $P(S|W)$.

The acoustic wave from the speaker's mouth will be further affected by the transmission channel from the mouth to the receiver to produce the received signal, X . The factors affecting the transmission channel will depend on the type of communication. In the simplest case of human speech communication this transmission channel will consist of the acoustic channel from the mouth to the ear, defined by the propagation properties of the room and the additive ambient noise. If the speech communication is conducted over the telephone, the channel will be combined of the acoustic channel from mouth to microphone (including effects of ambient noise on the transmitting end), the telephone transmission channel (including the transduction from acoustic to electric wave and back), and the acoustic channel on the receiving end. If we are looking at human-to-machine communication, the signal received by the speech recognizer will at least be affected by the acoustic channel from mouth to microphone and the characteristics of the microphone and A/D conversion. The mapping performed by the transmission channel is given by the probability $P(X|S)$.

At the receiving end, the human listener will try to reverse the production process to decode the original message. This process includes the analysis of the speech signal performed by the ear, which in turn produces electrical activity in the auditory nerve. The brain then applies knowledge of the language system in order to decode and comprehend the original message.

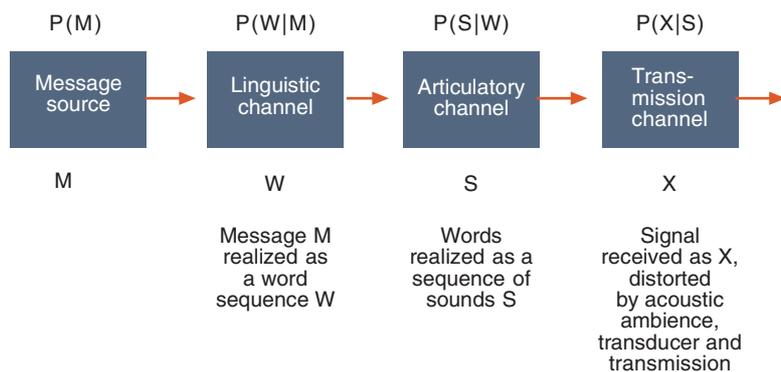
In order to better understand the rationales behind various approaches to speech recognition and synthesis, we will spend a little time looking at some fundamentals of speech production and perception before going into speech technology issues.

Speech Production

Speech is air pressure waves radiating from the mouth and nostrils of the speaker. The main components of the human speech production apparatus are the lungs, the glottis and the vocal tract. The driving source is air from the lungs. At the glottis, the vocal cords constrict the path from the lungs to the vocal tract. In voiced sounds, air pressure from the lungs build up behind the closed vocal cords, until they abruptly open to release a burst of air before closing again. The cycle repeats and produces a quasi-periodic sequence of excitation pulses. The inverse of the pulse period is called the fundamental frequency, which determines the perceived *pitch* of the speech signal. In unvoiced sounds, the vocal cords are open. The intonation of speech is determined by the variations of the fundamental frequency.

The excitation is filtered by the vocal tract to produce the sounds. The shape of the vocal tract, i.e. the position of the jaws, the opening of the lips, the shape of the tongue and the opening or closing of the velum will determine the frequency response of the vocal tract. Analysis of the vocal tract shows that the frequency response will typically be dominated by a small number of resonances, the *formants*. As we speak, we change the shape of the vocal tract, and thus the frequency response of the filter, in order to pro-

Figure 1 A communication-theoretic view of the speech communication chain (adapted from [16])



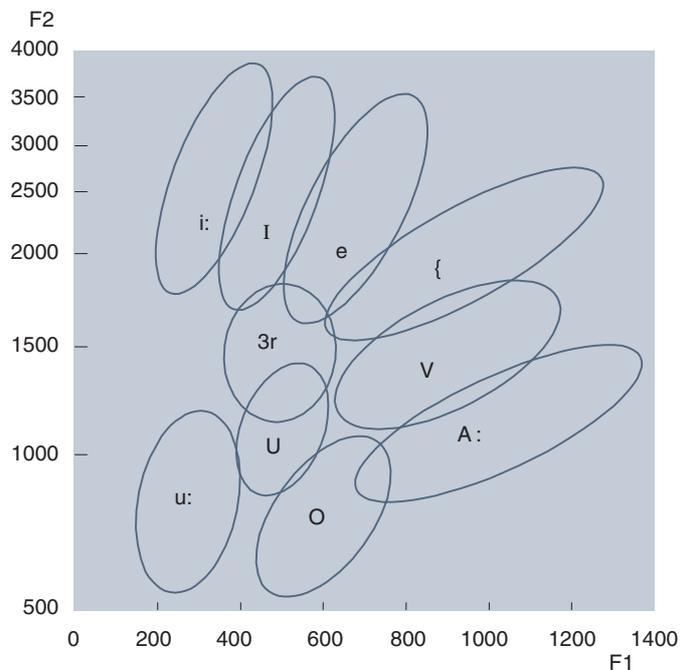


Figure 2 The vowel triangle showing the distribution of English vowels as a function of the first (F1) and second (F2) formants. The phoneme symbols are in SAMPA [35] notation

duce different sounds. The positions of the formants are the most significant factor in terms of human identification of speech sounds. Vowels can to a large extent be identified on the basis of the position of the two lowest formants, F1 and F2 (see Figure 2). However, the distribution of F1/F2 values for the different vowels has overlapping regions where the formant information is insufficient for making unambiguous decisions on vowel identity.

The smallest distinguishing unit of a language is the phoneme. The phoneme is an abstraction, covering a multitude of possible actual realizations, phones, of the sound. Different sounds are created by moving the articulators (e.g. tongue, lips, jaws). The articulators are physical entities with a mass, which means that their movement cannot be instantaneous. Thus, the realization of a phoneme will depend on the articulator positions of the preceding and the succeeding sounds. This phenomenon is called coarticulation. An accurate modeling of the coarticulation phenomena is vital for the performance of both speech recognizers and speech synthesizers.

Hearing and Perception

The human ear consists of three sections, the outer ear, the middle ear and the inner ear. The received air pressure wave travels through the outer ear and sets up a vibration at the eardrum. Vibration at the eardrum is picked up by the tiny bones of the middle ear which connect to the inner ear through the oval window of the cochlea. The cochlea is a liquid filled organ that connects directly to the auditory nerve. Vibrations at the oval window cause a displacement of the liquid along the basilar membrane of the

cochlea, the place of maximum displacement being dependent on the frequency content of the vibration. The displacement causes hair cells on the basilar membrane to be bent, resulting in neurological firings to the hearing nerve.

Human speech perception is dependent on the frequency of the stimulus. The cochlea performs a frequency analysis of the sound and decomposes it into spectral components, similar to a bank of overlapping bandpass filters where the bandwidth of each filter is increasing with the center frequency. It has been demonstrated that the bandwidth increase follows an exponential rule for frequencies above approximately 1 kHz and is close to linear below 1 kHz. This has prompted the use of the *mel* and *Bark* scales in speech analysis.

In terms of perception, the formants and the fundamental frequency are the factors that are the most important. Flanagan [13] reports results of a series of experiments aimed at determining the smallest deviations in isolated parameters for speech production that could be perceived by humans. The most sensitive parameter is the fundamental frequency where a deviation of 0.3–0.5 % could be detected. The frequencies and amplitudes of the two lowest formants were also parameters where small changes were easily detected. The formant bandwidths were less important, and humans seem to be fairly insensitive to deviations in the “spectral valleys”, between the formants.

Speech Technology

Speech technology comprises a number of different technologies. In the interest of space, this treatment will concentrate on automatic speech and speaker recognition and speech synthesis, excluding interesting areas such as speech coding, language identification and speech enhancement.

Speech Recognition

A speech recognizer is basically a device that upon the presentation of a speech utterance performs a predefined action. The predefined action can be to present a written transcription of the spoken utterance; to perform the action the utterance refers to (e.g. “Turn on the light”, “Open the last accessed document in Word”) or to extract the information in the spoken utterance that is relevant for a specific task and to interpret the meaning of that information (e.g.; if a user says “I need to go to the city to meet a friend, and I am running late. Can you tell me when the next bus from the University leaves?” the essential information a system for bus traffic information needs to extract is “Give departure time for next bus from University to city”).

A speech recognizer can be viewed as a classifier. The task of a classifier is to determine which class an observation belongs to. The number of classes is normally limited. For the speech recognizer the classes can be e.g. the words constituting the vocabulary of the recognizer. A typical measure of the quality of a classifier is the risk of misclassification. If we regard the speech recognizer as a word classifier and define the cost of erroneous classification uniformly to be 1 and the cost of correct recognition to 0, then the misclassification risk equals the word error rate, i.e. the expected percentage of words that will not be correctly recognized. The word error rate is the most widely used quality measure for speech recognizers.

The complexity of the task, and consequently the performance of the speech recognizer will depend on a number of factors. The most important are:

Vocabulary. The size of the vocabulary, i.e. the number of different words the recognizer is trained to recognize, will depend on the task the recognizer is designed for. Increasing the vocabulary size will increase the footprint of the recognizer, and it will also increase the risk of making errors. In addition, it is important to note that the content of the vocabulary can influence the performance of the system. Vocabularies containing many short, phonetically similar words are difficult to handle. This can be an important design factor for small vocabulary systems where it is possible to custom design the vocabulary content.

Speaking style. The speaking style can be isolated utterances or continuous speech. Isolated utterance systems are typically command and control applications, where the system only understands a limited number of commands. In continuous speech, the words of the vocabulary can be combined in any syntactically relevant manner to produce sentences. Continuous speech will typically allow an extremely high number of legal utterances, and will yield more complex systems than recognizers for isolated utterances. Large vocabulary continuous speech recognizers have traditionally required the user to speak in a carefully articulated manner, similar to reading. Spontaneous speech contains phenomena such as significant speaking rate variation, sloppy pronunciation, incomplete sentences and restarts, “uhms” and “ahs”, which most recognizers cannot handle. These artifacts are present e.g. in dictation, particularly if the user is not accustomed to dictating. Even more of these artifacts are present in conversations and dialogues, where problems like turn-taking and simultaneous speech are also present. The difficulties presented by speaking style is clearly illustrated when looking

at current state-of-the-art performance for conversational speech, which show word error rates around 35–40 %.

Speaker mode. The characteristics of speech depend on the speaker. This is partly due to physical differences in the vocal apparatus, e.g. differences in the length of the larynx, but dialects and accents will also contribute strongly to differences in the uttered speech. In order to achieve the best performance, a recognizer should ideally be trained on speech from the target speaker. This is impractical for most applications. Most modern speech recognizers are trained on speech from a large number of speakers, exposing the training algorithm to most normal variations in speech characteristics. For many applications this is sufficient to enable a satisfactory performance. In order to enhance the performance, the recognizer can adapt to the current speaker using whatever speech is available.

Speaker Recognition

In contrast to speech recognition, where the task is to identify *what* is spoken, the task of speaker recognition is to identify *who* is speaking. Speaker recognition can broadly be divided into two main application areas. The first is the task of verifying a speaker’s claimed or assumed identity, *speaker verification*, which is a binary (yes/no) classification problem. The second is *speaker identification*, which is to determine the identity of a speaker, either from a set of N known speakers (closed set), or from an unlimited set of speakers of which N are known to the system (open set). Closed set speaker identification is an N -class classification problem, while an open set speaker identifier can be viewed as an $(N+1)$ -class classifier where class $(N+1)$ represents all unknown speakers. As N increases, the probability of correct identification will decrease.

Speaker verification has received most interest, both because it is seen as having the highest potential for commercial applications (and is more reliable), and because many speaker identification applications can be recast to be implemented using speaker verification methods. Today, both speaker verification and speaker identification methods are based on very similar statistical paradigms. Thus, speaker verification will receive the greatest attention in this paper.

Speaker verification can be used for access control, which can include both physical access to restricted areas and logical access to various services. Examples of such services are telephone banking; telephone shopping; information access; travel services and computer account access. Speaker verification can also be applied to law enforcement by assessing the likelihood

that the voice in audio recordings made from e.g. telephone conversations is that of a specific person. It should however be noted that the use of speaker recognition is not admissible in a court of law in many countries. Another use of speaker recognition which lately has received considerable interest is speaker detection, or segmentation, which consists of identifying the current speaker in a multi-speaker audio recording such that segments of the recording can be automatically labeled with the identity of the speaker.

The intended application will determine the approaches to speaker recognition and to the reliability of the recognition. The main factors are:

Speech modality: Is the textual content of the speech known to the system or not? If the text is known, by requiring the user to speak a predefined or prompted phrase, system reliability can be very high. Such systems are termed text dependent. In text independent systems, the textual content is not known. The phonetic content of the input speech is thus also unknown, and the system cannot utilize knowledge about the speaker specific phone realizations, resulting in lower reliability.

Speech quality: The quality of the recording will have a strong impact on the performance of speaker recognition systems. Channel and ambient noise in an application where the user is calling the system from a mobile phone will cause degradation in the system performance relative to an application where the user is speaking directly into the microphone in a quiet environment. Similarly, forensic speaker recognition based on a (hidden) omnidirectional surveillance microphone in a reverberant room is far more difficult than recognition from a recording of a fixed line telephone conversation.

Amount of speech: Generally, a speaker recognition system will perform better the more speech can be made available for recognition. In many applications there will obviously be a trade-off between reliable identification and user convenience. A speaker recognition system requires speech data to train the speaker models, i.e. to “learn” the characteristics of the speakers. The amount of speech data available for training will have a strong influence on system performance.

Evaluation of system performance will be dependent on the boundary conditions defined by the application. For speaker verification systems there are two types of errors, *false acceptance*, i.e. that the identity claim/assumption is accepted although the speaker is an imposter, and *false rejection*, i.e. that the identity claim is rejected although the voice belongs to the claimed speaker. The severity of the two error

types will vary depending on the application, and deciding the relative costs is an important design criterion. False acceptance constitutes a security risk, while false rejections mainly impact on user convenience. A system designed for a high security application will thus attempt to minimize the false acceptances at the cost of accepting a higher rate of false rejections.

Speech Synthesis

Speech synthesis is generally the act of making the computer generate a spoken message from a textual concept. If the message vocabulary is small, the spoken message can be generated either by simple playback of pre-recorded messages or by *phrase concatenation*. During system design a library of pre-recorded phrases, i.e. parts of sentences, are recorded. For a talking clock a library consisting of the phrase “The time is now” and pronunciations of the numbers 0–20, 30, 40 and 50 would suffice to enable the computer to tell the time in hours and minutes. For example, at 21:37 hours, the clock can say the time by concatenating the carrier phrase “The time is now” with the words “twenty”; “one”; “thirty” and “seven”. In applications where the output messages are syntactically restricted and the number of messages is limited, a properly designed phrase concatenation system gives very high quality at a low cost. However, if it is desirable to change the message vocabulary a complete redesign is usually needed – unless the speaker that contributed the original recordings is available and the recording environment can be fairly closely reconstructed.

Although phrase concatenation has been widely used in applications with limited message vocabularies, the term “speech synthesis” is usually associated with text-to-speech synthesis (TTS). TTS implies that any text can be rendered as speech by the speech synthesizer. An important part of a TTS system will of course be the actual synthesizer, i.e. the conversion of a (textual) string of symbols to speech. But the TTS system must also perform a linguistic analysis of the input orthographic text in order to extract information about pronunciation, stress and timing, estimate where the emphasis should be put in a sentence, disambiguate words and phrases when multiple interpretations will yield different pronunciations, etc.

The quality of a TTS system will depend on both the linguistic processing and the speech synthesis. The basic linguistic processing will for instance ensure that numbers and abbreviations are correctly interpreted and pronounced and is thus an important factor in the overall quality assessment. However, the most widespread assessment strategies for TTS emphasize *naturalness* and *intelligibility*. Intelligibility is

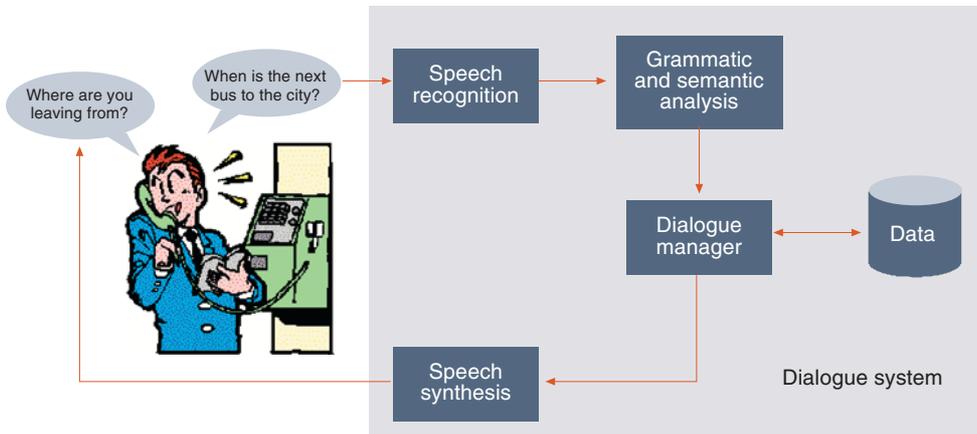


Figure 3 A spoken dialogue system for bus traffic information

often measured by the capability of human listeners to correctly identify the phonetic content of short, nonsense words or the words of semantically unpredictable sentences. Naturalness on the other hand is typically measured using techniques developed for evaluating speech compression algorithms, such as a subjective evaluation of quality by paired comparison or mean opinion score (MOS).

Spoken Dialogue Systems

In a spoken dialogue system, speech recognition is used to convert spoken requests and information to a symbolic form understandable by the information retrieval module. The speech recognizer can be used in conjunction with a semantic analysis module which attempts to extract the semantically meaningful part of the utterance, or the semantic processing can be integrated in the recognizer. Speech synthesis is employed to prompt the user for information and to convey the information the user has requested. In addition to the speech I/O modules, a vital part of a spoken dialogue system is the dialogue manager. The dialogue manager controls the dialogue, keeps tabs on the dialogue flow so that the system is not lost if the dialogue should not follow a linear path, performs recovery from errors made by the recognizer or the user, and acts as an interface between the information retrieval module and the speech modules. An example dialogue system is depicted in Figure 3.

The Past and Present

This section will attempt to give a brief overview of the most important developments in speech recognition and synthesis up till today. By necessity, the presentation will be rather superficial, leaving out many significant contributions to the development of speech technology.

Speech Synthesis

Early efforts at constructing speaking machines were undertaken as early as the second half of the 18th century. Wolfgang von Kempelen constructed a "speaking machine" [18] which was

an ingenious mechanical approximation of the human speech apparatus, using a bellows to simulate the function of the lungs, a reed to approximate the vocal cords and a flexible rubber tube could be manipulated to shape the "vocal tract" to produce the correct sound.

An electronic analogue to von Kempelen's machine was demonstrated by Homer Dudley at the 1939 World Fair [8]. In Dudley's VODER, the user could manipulate the resonances of the vocal tract, thereby affecting the sound characteristics, and the voicing and pitch could also be controlled by levers and pedals. Making the VODER speak was like playing an instrument, and although both the VODER and von Kempelen's speaking machine demonstrated that it was possible to make machines produce fairly intelligible speech, it could not be done automatically. The first attempt at automating the procedure was done in the early 1950s, when the pattern playback machine [6] was designed to produce a sound pattern that followed a spectrogram.

The early efforts at producing speaking machines focused more on the sound production than on text analysis. In fact, this has been the case up till today. Yet, the quality of the linguistic analysis of the text, including text normalization, homograph disambiguation, stress analysis and grapheme-to-phoneme conversion constitute will define the upper bound on the achievable quality of the complete TTS system. Still, here we will only remark that the text analysis is developing from a traditional, rule based approach to include more and more data driven approaches, e.g. by exploiting statistical methods for inferring general rules from exposure to examples. The remainder of this section will treat sound production.

From the early days there were two competing paradigms for speech synthesis. One approach aimed at modeling the articulators of the speech production mechanism in detail, *articulatory synthesis*. The other approach was targeted at

modeling the speech signal itself, through a source-filter model [10]. This is often also termed *terminal-analog* synthesis [12]. Modeling the signal is intrinsically simpler than modeling the articulators, and the terminal-analog synthesizers were dominant, with formant and linear predictive synthesizers being prominent examples. Articulatory synthesis still remains an active field of research.

The formant synthesizer is based on *synthesis-by-rule*. In a synthesis-by-rule synthesizer, rules for parameter values and trajectories corresponding to explicit phonetic phenomena are used to drive the synthesizer. The parameters are typically divided into source and filter parameters, e.g. voicing, durations, formant frequencies and bandwidths. The rules are extracted from parameterized speech data, and the extraction and refinement of the rules can be a grueling task. Perhaps the most prominent example of formant synthesizers is Dennis Klatt's Klattalk [19] which was later commercialized as DECTalk [5]. The strength of rule-based synthesis is that in principle, the rules can be applied to e.g. change the speaker characteristics and the speaking style without the necessity of a major redesign of the synthesizer. The clear disadvantage is that even in the best systems, the naturalness of the synthetic voice is limited.

The linear predictive synthesizer was the first example of *concatenative speech synthesis* which deviates from the synthesis-by-rule paradigm in that the synthesizer possesses a very limited explicit knowledge; most is embedded in the actual segments to be concatenated [9]. In a linear predictive synthesizer the filter parameters are generated from analyses of actual filter trajectories of speech segments. Since coarticulatory influences are smaller at the center of a phoneme, and the transitions between phonemes are perceptually crucial, the diphone was proposed as the basic synthesis unit. The diphone is a unit which stretches from the middle of one phone to the middle of the succeeding phone. The linear predictive model of speech is a simple source-filter model and lends itself readily to prosodic modification by altering the excitation (for pitch modification) or by interpolating the filter parameters (for modifying segment durations). Yet, the model has intrinsic weaknesses in terms of the achievable quality of the synthetic speech. The speech quality can be improved by introducing better production models, e.g. by the use of Multi-pulse excited LPC [1]. However, improving the production model comes at the cost of making prosodic control more difficult.

The next major step forward was the introduction of the *Pitch-Synchronous Overlap Add*

(PSOLA) technique [23]. The PSOLA approach allows manipulating the prosodic features of stored diphone waveforms, and the concatenation of the prosodically altered segments can be performed using the same paradigm. Thus, any desired sequence of phones exhibiting a desired prosody can in principle be generated from a limited diphone inventory. In practice there is a limit to what extent the prosody can be manipulated before it results in audible distortion. Also, the technique does not directly solve problems associated with spectral or phase discontinuities at the concatenation points. Nevertheless, the PSOLA technique was a quantum leap in terms of improving the naturalness of synthetic speech.

Diphone based PSOLA techniques depend on predicted prosodic characteristics, and the speech quality is vulnerable to discontinuities at the joints of the diphone segments. *Unit selection synthesis* (see e.g. [24, 4, 3]) is designed to avoid these disadvantages by eliminating, or greatly reducing, the need for prosodic manipulation of the stored waveforms. Instead of having a carefully designed diphone inventory with only a single, neutral realization of each diphone, unit selection synthesis is based on the availability of a large speech database, containing most of the natural prosodic variation. The basic idea is that if a desired phonetic and prosodic context can be found in a database of natural speech, nothing can be more natural sounding than that. The principle of unit selection synthesis is that the (appropriately processed and labeled) database is searched for a sequence of units that best matches the sequence that is predicted by the text analysis part of the TTS engine. The search of the database can be performed using dynamic programming. The cost of using a specific unit from the database is combined of two parts, the target cost, which is a measure of the match between the database unit and the desired unit, and the concatenation cost, which is a measure of the distortion that will arise by concatenating the unit with a preceding database unit. Letting the concatenation cost be zero for database units that are spoken in succession leads the search to favor unit sequences that occur naturally. This reduces the amount of spectral discontinuities and tends to imitate prosodic structures found in the database. Unit selection synthesis can produce very natural sounding speech, but can also fail miserably if the database does not contain good matches. Critics of the technique emphasize the data coverage problem as the main problem: due to the enormous variations in speech, it is impossible to collect (and use) a database that contains all possible phonetic and prosodic variation. However, allowing limited, high quality prosodic manipulation of the database units may reduce the validity of this criticism.

Speech Recognition

The first speech recognizer of sorts was demonstrated in the late 1920s. *Radio Rex* was a mechanical toy dog that would jump out of his house when his name was spoken. The action was basically triggered by the level of sound energy around 500 Hz, which is the frequency band around the first formant of the vowel 'e' (which implies that Rex would jump into action also as the result of other acoustic stimuli than his name). The first real speech recognizer was a device capable of recognizing the digits 0–9, developed by researchers at Bell Labs in the early 1950s [7]. This device identified the digits on the basis of a crude estimate of the formants of the vowels contained in the digits.

With the advent of the digital computer more sophisticated methods were made possible. In speech recognition the two most important contributions to the progress were the introduction of dynamic programming techniques for time-scale adjustments when comparing patterns and the introduction of robust and efficient methods for spectral estimation.

Temporal variation in speech tends to be non-linear. In order to compare two patterns of different length, but where the ratio of the durations of the speech sounds in the two patterns is not constant, the use of dynamic programming for time alignment was introduced ([17, 25]). *Dynamic Time Warping* was a dominant technique for speech recognition until the mid-1980s, and systems using this simple method for matching time patterns can still be found. Still, most important was the introduction of dynamic programming methods to speech processing. Dynamic programming is a very powerful tool and is extensively used in current speech processing algorithms (see e.g. [21]).

The speech waveform itself exhibits too much variation to be well suited for speech recognition. Even in the early systems it was realized that the spectral domain is well suited for distinguishing between sounds. The introduction of short-term spectral estimation methods and corresponding distortion measures that were reasonably robust to speaker variations and which corresponded well with hearing constituted a major step forward. Several of these innovations came from outside the speech recognition community (e.g. linear prediction analysis, cepstral representation of the spectrum, distortion measures), but were readily applicable to ASR.

The early systems for speech recognition tended to adhere to two paradigms: pattern matching and/or acoustic-phonetics. Pattern matching systems based on Dynamic Time Warping tried to match incoming speech patterns with stored ref-

erence patterns, taking into account the non-linear time variation exhibited. The reference patterns were typically word-length, making it difficult to construct large vocabulary systems. Systems based on acoustic-phonetics originated in the study of the properties of the speech sounds, the *phonemes*. The phonemes can be classified depending on how and where in the vocal apparatus they are generated; i.e. the articulatory configuration typically observed for each sound. There are two main problems with the acoustic-phonetic approach: the difficulty of reliably estimating the articulatory features from a speech sample, and the insufficient description of the variability in real, spontaneous speech, often contaminated by noise. Thus, these approaches could be successful for limited vocabulary systems with carefully articulated speech, but were not adequate for solving the general problem of speech recognition.

A major step forward came with the introduction of statistical approaches for speech recognition. Hidden Markov Models (HMM) were proposed for speech recognition in the mid-70s ([2, 15]) and were within a decade to become the de facto standard approach to speech recognition. The success of the statistical approach, and HMM in particular, can be attributed to the following main factors:

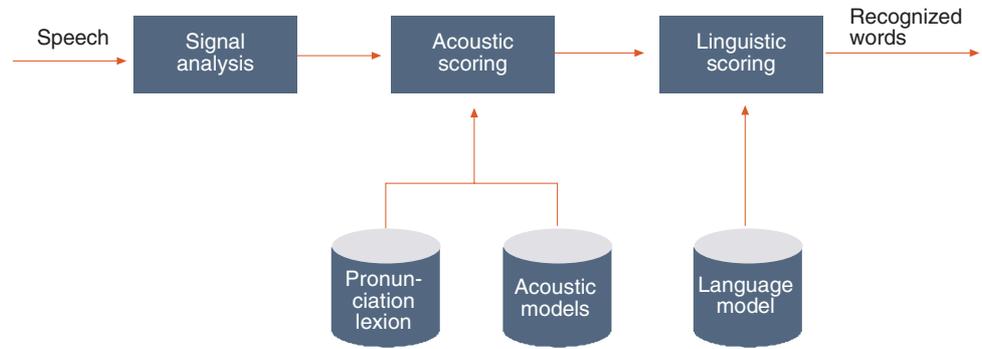
- 1 The statistical approach has the capability of modeling the variation found in normal speech.
- 2 The emergence of large databases made it possible to obtain reasonable parameter estimates for the complex statistical models needed to describe this variation.
- 3 The statistical framework is applicable to modeling the probabilistic mappings in the speech communication chain, in principle enabling a unified approach to decoding the underlying message.

Most research has concentrated on modeling the acoustic and linguistic mappings in Figure 1. The optimal classifier for determining the word sequence, $\mathbf{W} = (w_1, w_2, \dots, w_N)$ that is the source of an acoustic observation, \mathbf{X} , will select the sequence \mathbf{W}' that maximizes the a posteriori probability, i.e.

$$\begin{aligned} \mathbf{W}' &= \arg \max_{\mathbf{W}} \{P(\mathbf{W}|\mathbf{X})\} \\ &= \arg \max_{\mathbf{W}} \left\{ \frac{P(\mathbf{X}|\mathbf{W}) \cdot P(\mathbf{W})}{P(\mathbf{X})} \right\} \end{aligned} \quad (1)$$

Since the maximization is independent of $P(\mathbf{X})$, this term can be neglected. Thus, the choice of the optimal word sequence is dependent on two

Figure 4 Statistical speech recognizer



terms, the acoustic likelihood, $P(\mathbf{X}|\mathbf{W})$ and the language model $P(\mathbf{W})$. A simplified block diagram of a statistically based speech recognizer is shown in Figure 4.

The task of the statistical recognizer is thus to find the most likely word sequence for an observed spoken utterance, represented by a sequence of feature vectors produced by the signal analysis. In order to do this, statistical models for the various speech sounds must be estimated in a training phase. Normally, several models are produced for each speech sound, dependent on the phonetic context surrounding the target sound. This is done in order to model coarticulation effects. In order to describe the words of the recognizer vocabulary, a pronunciation lexicon defining the words in terms of phone sequences is required. Also, a language model must exist. This will in most cases be a statistical N-gram model, i.e. a model that estimates the probability of an N-tuple of words. For example, a trigram language model restricts the language modeling by assuming that the probability of a target word is only dependent on its two most recent predecessors, i.e. $P(w_n | w_{n-1} w_{n-2} \dots w_2 w_1) = P(w_n | w_{n-1} w_{n-2})$. Then, the probability of a word sequence (e.g. a sentence) $W = \{w_1, w_2, \dots, w_N\}$ can be found as

$$P(w_1 w_2 \dots w_N) \rightarrow P(w_1 w_2 \dots w_N) \quad (2)$$

Other notable advances in automatic recognition include the use of a perceptually based warping of the frequency axis when performing the signal analysis, and the use of a decorrelating (cosine) transform for producing mel-cepstrum feature vectors as the input to the statistical classifier. Mel-cepstrum feature vectors augmented by estimates of their first and second order time derivatives are standard features of current speech recognizers.

Speaker Recognition

The development of speaker recognition technology has to a large extent paralleled the evolution of speech recognition. As for a speech recognition system, an operational speaker recognition system will consist of an analysis

stage and a pattern matching stage. The analysis is aimed at extracting features that are robust and that are able to represent the speaker characteristics. Identity information that is embedded in the speech signal includes information on the physiology of the vocal tract, suprasegmental information like pitch and stress patterns and higher level information such as pronunciation, word and phrase frequencies and syntax. Suprasegmental and higher level information are relatively easy to mimic (these are features typically used by human mimics), and are not well suited for most applications of speaker recognition. The features used for speaker recognition are basically the same as used for speaker independent speech recognition, namely spectral features containing information on the vocal tract shape. This is intriguing, as the aim of the feature extraction in speaker recognition is to extract parameters that consistently and robustly emphasize the differences between speakers, while a speech recognizer front end should preserve the phonetic characteristics and aim to minimize the influence of speaker variations. However, both the main phonetic and the speaker dependent characteristics are dependent on the shape of the vocal tract, which is efficiently described by the smoothed spectral envelope that can be represented by e.g. cepstral or mel-cepstral parameters. Interestingly, the use of dynamic parameters (time derivatives of e.g. mel-cepstral parameters) was first proposed for speaker verification [27, 28] but is now common in both speech and speaker recognition systems.

The pattern matching stage compares the extracted features to stored models for each speaker known to the system in order to find the best match. Early systems used simple template matching techniques. More advanced template matching methods employing statistical feature averaging were introduced for text-independent speaker recognition in the 1970s. Here, a template is created for each speaker by averaging the feature vectors over the training speech. In recognition, a distance measure evaluating the deviation between the test speech and the stored template is evaluated and constitutes the basis for the decision. Dynamic Time Warping, which

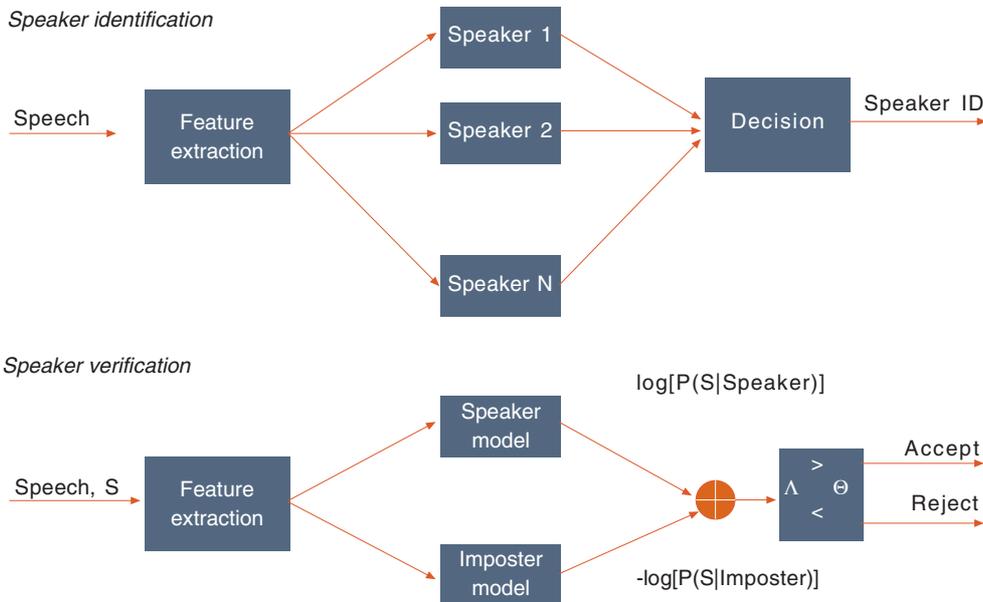


Figure 5 Speaker identification and verification

was introduced for speech recognition around 1970 was quickly adapted to speaker recognition, particularly for text-dependent applications. Here, templates representing words or utterances are created during training, and the test speech is compared and evaluated against these reference patterns.

The predominant approach to speaker recognition over the past 15 years has been based on probabilistic modeling. Nearest Neighbor modeling, which is a non-parametric approach, is based on representing each speaker's acoustic space by a collection of feature vectors. The collection can consist of all feature vectors in the training speech [29] or a vector quantization codebook designed from the training speech [30]. The distance between an input feature vector and its nearest neighbor is a measure of the probability of the assumed speaker, and is the basis for calculating the speaker score, which can be viewed as an estimate of the probability that the test speech emanates from a given speaker.

Yet, the parametric approach of Hidden Markov modeling has lately been the most successful approach to speaker recognition, as it has been for speech recognition. The basic HMM paradigm can be applied to both text-dependent and text-independent speaker recognition. In text-dependent speaker recognition, speaker models are created as word models or phone models, depending on the application. Using phone models, the speaker models can be used to verify any spoken word sequence. Word (or phrase) models restrict the application to a predefined vocabulary. For text-independent applications, a single-state HMM is used to model each speaker's acoustic space. This approach, usually referred

to as Gaussian Mixture Models (GMMs) [31], can also be viewed as modeling a speaker as a probabilistic source with a probability distribution that is a weighted sum of multivariate Gaussian densities. Recognition will then basically consist of estimating the likelihood of the speech being produced by a speaker and making a decision based on the estimated likelihoods.

The likelihood score obtained using HMM approaches can exhibit significant variation due to factors that are not related to the speaker, such as noise; microphone, channel and text variability. These effects can be reduced by processing the extracted feature by e.g. cepstral mean normalization [34]. An efficient method for reducing these effects are imposter models, which are used to normalize the likelihood score by letting the input speech be scored against both the target speaker and the imposter models [32, 33].

In speaker verification systems, it is vital to minimize the risk of accepting imposters. The algorithms for speaker verification take this into account by basing the decision on features that describe the vocal tract, a feature that is hard to mimic. However, in many applications one needs to consider the possibility of intruders using analog or digital recordings of a speaker. This can successfully break the system security if it is text-independent, or if fixed pass-phrases are used. Because of this, most current applications use prompted phrases. The system will generate a sentence which the user is asked to utter. In this way, the verification can be text-dependent (as the system knows which sentence to expect) but since the same sequence of words are never repeated; recordings cannot be used to defeat the system.

Brief Summary of the State-of-the-art

Speech synthesizers have during the last decade moved from being fairly intelligible but not very natural sounding to being (at best) quite natural sounding through the application of PSOLA-techniques and unit selection synthesis. For speech synthesis within a specific task domain, e.g. reading of news reports, unit selection synthesis currently produces (on average) quite natural sounding synthetic speech, although the prosodic variation is more limited than in human speech. For general reading machines, where it is desirable to produce a multitude of speaking styles, including the ability to express emotions, unit selection synthesis is currently not applicable. It is still an open question whether this will be attainable using corpus based speech synthesis.

Two of the major problems associated with unit selection synthesis is the footprint of the system and the complexity of creating new voices. The size of the database used in the synthesis will reflect on both memory requirement of the synthesizer and the computational requirements. Database design and organization are important issues if unit selection synthesis shall be available e.g. for handheld devices. Creating a new voice will include the recording and annotation of a large speech database. In particular, the processing and annotation of the data is time consuming and costly if done manually. Recent efforts at automating the process [26] have shown very promising results, and can lead to great savings in time and cost of creating new synthetic voices.

The state-of-the-art in speech recognition is mainly dependent on speaking style and the noise environment. For clean speech, i.e. when the effects of ambient or transmission noise is negligible, the performance of current speech recognizers is adequate for a large number of tasks as long as the input speech is fairly well articulated and fluent. Dictation systems for limited domains, such as radiology, where the amount of variation in syntax and vocabulary is limited, and the speakers are used to dictation, have been well received. Good performance for general speech-to-text remains a more elusive goal. When it comes to spontaneous and conversational speech, the performance of current speech recognizers is far below what is acceptable for most applications. Systems for automatically transcribing spontaneous monologues or natural conversations are beyond the capabilities of today's technology.

One of the main problems for current speech recognizers is the performance in the presence of noise. Humans have little problems with understanding noisy speech, even at high noise

levels, but speech technology has yet to find good methods for handling the effects of additive or convolutional noise. Many approaches have been investigated, but the problem is far from solved. Even relatively simple tasks, such as isolated word recognition for command and control applications, can be severely affected if the noise conditions do not match the environment for which the recognizer was trained.

Systems for spoken dialog, e.g. for automated information retrieval, have been successfully deployed. The complexity and performance of these systems depend on the dialog structure adopted. System driven dialog systems for restricted tasks can be well performing, particularly if techniques such as word spotting or concept spotting is employed, allowing words not relevant for the task to be ignored or even mis-recognized. Mixed initiative, or even user initiative systems such as AT&T's "How may I help you?" system for routing telephone calls [14], are not quite as readily deployed, although very encouraging results are obtained.

The Future

One of the major trends that can be observed from the progress in speech technology over the past decades is that there is a convergence of the techniques employed in the various disciplines. It is a clear tendency to rely on data driven methods, using statistical tools for modeling. In speech synthesis this is exemplified by the corpus based unit selection synthesis, the HMM-based methods for segmenting and labeling the speech databases, and also statistical methods for linguistic analysis.

In speech recognition, statistics has ruled the ground for the past 15 years, dominating both the acoustic modeling and the language and grammar. Even for semantic processing, statistical methods have shown great promise. On the other hand, speech recognition has over the same period of time moved from being a more or less pure signal processing discipline to requiring the use of explicit linguistic knowledge. This is of course connected to the evolution of the task complexity, which in many cases today requires the application of knowledge of linguistic, phonetic and discourse structures.

One interesting recent development is the interest in finite state automata for speech recognition. Although finite state networks have been a useful representation of both simple grammars as well as the state networks used for HMM decoding, work at AT&T has introduced a weighted finite state transducer (FST) formalism for describing the entire decoding process, including the acoustic models, the vocabulary and pronunciation lexicon, context dependency

and language model [22]. The immediate advantage of the FST formulation is that it allows for designing faster and more efficient decoders. However, the formalism can also give new insights into speech recognition and allows for a very interesting connection to the use of finite state machine formalisms in text-to-speech synthesis and in computational linguistics.

In spite of the convergence of techniques, there is still no “Grand unifying theory of speech” that can guide the way to better speech technology. We still lack some of the fundamental understanding of speech and language communication that can produce truly speech *understanding* systems, generation of natural sounding (emotional) synthetic speech from a concept, and *conversational* machines. We are currently making up for some of the lack of theory by employing statistics, with some degree of success.

Conclusions

Speech technology has come a long way from its primitive origins, and the technology is currently at a stage where a significant number of useful and profitable applications can be made. Still, it is important to be aware of the limitations of the current technology, both for designing good and user-friendly systems today, and for aiming to solve the challenges of tomorrow. Speech technology is not mono-disciplinary, and in order to be able to create a speech-enabled machine that can pass the Turing test, the collaborative efforts of scientists from many areas are needed.

Among the immediate challenges are: how to design speech recognizers that are robust to speaker accents and dialects as well as to noise contamination, how to create speech synthesizers that are able to convey emotions and can adopt speaking styles appropriate for any given text, how to deal with multi-linguality, how to attack the problem of recognizing spontaneous speech, just to name a few. Many problems have been solved, but speech technology research will not lack challenges in the years to come.

References

- 1 Atal, B S, Remde, J R. A new model for LPC excitation for producing natural-sounding speech at low bit rates. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Paris, 614–617, 1982.
- 2 Baker, J K. The DRAGON system – An overview. *IEEE Trans. Acoustics, Speech and Signal Processing*, 23 (1), 24–29, 1975.
- 3 Beutnagel, M et al. *The AT&T NextGen TTS system*. Joint meeting of ASA, EAA and DAGA, Berlin, 1999.
- 4 Black, A, Campbell, N. Optimising selection of units from speech databases for concatenative synthesis. *Proc. Eurospeech '95*, Madrid, 581–584, 1995.
- 5 Bruckert, E, Minow, M, Tetschner, W. Three-Tiered Software and VLSI Aid Developmental System to Read Text Aloud. *Electronics*, 21 April 1983.
- 6 Cooper, F S et al. Some experiments on the perception of synthetic speech sounds. *J. Acoust. Soc. Am.*, 24, 739–764, 1952.
- 7 Davis, K H, Biddulph, R, Balashek, S. Automatic Recognition of Spoken Digits. *Journal of the Acoustical Society of America*, 24 (6), 637–64, 1952.
- 8 Dudley, H W, Riesz, R R, Watkins, S A. A Synthetic Speaker. *Journal of the Franklin Institute*, 227, 739–764, 1939.
- 9 Dutoit, T. *An Introduction to Speech Synthesis*. Dordrecht, Kluwer Academic Publishers, 1997.
- 10 Fant, G. *Acoustic theory of speech production*. The Hague, Mouton and co., 1960.
- 11 Ferguson, J. *Hidden Markov Models for Speech*. Princeton, NJ, IDA, 1980.
- 12 Flanagan, J L. Notes on the design of Terminal-Analog speech synthesizers. *J. Acoust. Soc. Am.*, 29, 306–310, 1957.
- 13 Flanagan, J L. *Speech analysis, synthesis and perception, 2nd Edition*. Berlin, Springer-Verlag, 1972.
- 14 Gorin, A, Riccardi, G, Wright, J H. How May I Help You. *Speech Communication*, 23, 113–127, 1997.
- 15 Jelinek, F, Bahl, L R, Mercer, R L. Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Trans. Information Theory*, IT-21, 250–256, 1975.
- 16 Juang, B-H, Furui, S. Automatic recognition and understanding of spoken language – a first step toward natural human-machine communication. *Proc. IEEE*, 88 (8), 1142–1165, 2000.
- 17 Viyntuk, T K. Speech discrimination by dynamic programming. *Kibernetika*, 4, 81–88, Jan.-Feb. 1968.

- 18 von Kempelen, W. *Mechanismus der menschlichen Sprache nebst Beschreibung einer sprechenden Maschine*, 1791.
- 19 Klatt, D H. The Klattalk text-to-speech system. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Paris, 1589–1592, 1982.
- 20 Klatt, D H. Review of text-to-speech conversion for English. *J. Acoust. Soc. Am.*, 92 (6), 737–793, 1987.
- 21 Lee, C-H. Applications of dynamic programming to speech and language processing. *AT&T Technical Journal*, 68 (3), 114–130, 1989.
- 22 Mohri, M, Pereira, F, Riley, M. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16, 69–88, 2002.
- 23 Moulines, E, Charpentier, F. Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9 (5-6), 453–467, 1990.
- 24 Sagisaka, Y et al. ATR – v-TALK speech synthesis system. *Proc. ICSLP'92*, Banff, 483–486, 1992.
- 25 Sakoe, H, Chiba, S. A Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Trans. on ASSP*, 26 (27), 43–49, 1978.
- 26 Syrdal, A et al. Corpus-based techniques in the AT&T NextGen synthesis system. *Proc. ICSLP 2000*, Beijing, 2000.
- 27 Sagayama, S, Itakura, F. On Individuality in a Dynamic Measure of Speech. *Proc. ASJ Spring Conf. 1979*, 3-2-7, 589–590, June 1979.
- 28 Furui, S. Cepstral Analysis Technique for Automatic Speaker Verification. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 29, 254–272, 1981.
- 29 Higgins, A L, Bahler, L G, Porter, J E. Voice identification using nearest-neighbor distance measure. *Proc. ICASSP'93*, Minneapolis, II, 375–378, 1993
- 30 Soong, F K et al. A vector quantization approach to speaker recognition. *Proc. ICASSP'85*, Tokyo, 387–390, 1985.
- 31 Reynolds, D A, Rose, R C. Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE Trans. on Speech and Audio Processing*, 3, 72–83, 1995.
- 32 Higgins, A L, Bahler, L G, Porter, J E. Speaker verification using randomized phrase prompting. *Digital Signal Processing*, 1, 89–106, 1991.
- 33 Carey, M, Parris, E, Bridle, J. A speaker verification system using Alphanets. *Proc. ICASSP'91*, 397–400, 1991.
- 34 Atal, B. Automatic recognition of speakers from their voices. *Proceedings of the IEEE*, 64, 460–475, April 1976.
- 35 *SAMPA Computer Readable Phonetic Alphabet*. July 29, 2003 [online] – URL: <http://www.phon.ucl.ac.uk/home/sampa>

Computational Linguistics Methods for Machine Translation in Telecommunication Applications?

TORBJØRN NORDGÅRD



Torbjørn Nordgård (44) is professor of computational linguistics at the Norwegian University of Science and Technology (NTNU). He obtained his Master in Linguistics from NTNU (1985) and received his PhD in Computational Linguistics from the University of Bergen in 1991. Research activities include automatic text analysis (parsing, POS-tagging), formal syntax, computational lexicography, text generation, machine translation, statistical language modelling and text proofing tools for dyslectics. Nordgård is also CEO of LingIT AS (www.lingit.com).

torbjorn@hf.ntnu.no

This paper focuses on certain properties and research issues in automatic machine translation and their potential relevance for telecommunication applications. Machine translation (MT) is chosen as an example area for two reasons. On the one hand MT involves central computational linguistics (CL) activities like natural language parsing, semantic analysis, discourse analysis and text generation. Secondly, automatic translation with speech interfaces appears to be one of the most challenging and interesting research domains in natural language technology. This paper will give a brief overview of relevant aspects of MT and CL and discuss possibilities and limitations for telecom applications.

1 What is Machine Translation?

Machine Translation (MT) is transfer of text from one language to another by use of computers. The need for MT has been articulated for more than 50 years by industry and governmental institutions. After World War II it was assumed that translation was more or less the same as “code cracking” as it was called in military intelligence services. So, translating Russian texts into readable English was assumed to be more or less the same as interpretation of coded German messages to submarines into readable English. In retrospect, it is fair to say that the highly optimistic opinion in the postwar years was totally misguided, and nowadays it is replaced by a more modest and realistic view.

2 Why is MT Difficult?

In the mid sixties it became clear that the “simple” methods used in the first generation MT systems did not deliver translations with reasonable quality. The problem with this approach, and indeed every MT approach, is, somewhat simplified, that syntactic variation (word order variation) and ambiguity make the “mechanic” translation process far more complex than “code cracking”. In addition, good translations involve interpretation and understanding of the source texts prior to text production in the target language. Fluent speakers of English know whether a phrase or sentence is an idiom or a collocation, e.g. “kick the bucket”, ... , “walk her home”. It is of course harder for a computer to decide whether phrases are to be interpreted literally or not. Generally, as every professional translator and most people with academic background know, translation is indeed difficult, and often the problem boils down to choosing the most appropriate translation rather than a translation which seems to work. Given the complexity of the translation problem for humans, it is not surprising that computers face problems when they are set to translate between natural languages.

3 Is MT Possible?

It is quite natural to ask whether MT is possible at all. If computers in general are expected to produce translations just as good as human translations, the answer is clearly no. But if the goal is more limited, for instance that computers translate in very restricted textual domains, that an automatic system should propose translations which human translators can choose from, or that the expected quality is “raw translations” where errors are allowed to occur, then MT is here already. Consider for instance the Babelfish translation system which is part of the search engine Alta Vista, where users can ask for translations from e.g. German to English.

4 Rule-Based MT

Most existing MT systems are rule-based. That is to say, the knowledge of the systems is formulated by means of carefully designed rules which describe grammatical properties of the source and target languages, word pairings between the languages in question and “transfer” rules which couple expressions between the relevant languages. The exact nature of these rules vary considerably. Modern experimental systems utilize powerful description languages whereby it is possible to “decorate” phrase structure rules, as they are known from e.g. compiler theory, with so-called unification equations. By doing so it is possible to capture complex and subtle grammatical details of the languages by applying a manageable number of rules and general principles.

The automatic translation process consists at least of the following modules:

(Preprocessing) -> Analysis -> Transfer
-> Generation -> (Postprocessing)

The first phase takes care of various formatting issues so that the analysis component can be reasonably clean, i.e. without grammatically irrelevant details. Analysis consists of some sort of parsing, which is automatic analysis of text from the source language. The parsing result is some kind of linguistic structure. The details of the

parsing operations are dependent on which transfer strategies are used (we return to transfer strategies below). After the transfer operations are completed the system is ready to produce sentences in the target language on the basis of the output from the transfer component. If the generation step gives a set of sentences, some postprocessing device must pick the most likely candidate, or a human will have to choose the best alternative.

4.1 Analysis

As just mentioned, analysis involves parsing of the source text. Modern systems use phrase structure rules with unification equations. We illustrate this with a simplified, but realistic example from English:

```
S -> NP VP
  S:properties = VP:properties
  VP:properties:subject = NP:properties
  VP:properties:head:number =
  VP:properties:subject:number
```

The first line of the rule says that a sentence (S) consists of a noun phrase (NP) and a verb phrase (VP). So, in a simple sentence like “Saddam dislikes George” the words “dislikes George” constitute the verb phrase (or “predicate” if one prefers this notion), and “Saddam” is the noun phrase (NP) in the rule. The first equation in the rule says that the properties of the sentence (S) are the same as the properties of the VP. The second equation states that the subject of the VP, and hence the subject of the sentence, since S and VP share their properties, is to be understood as the properties of the NP. And finally, the rule requires that the grammatical number of the subject must be the same as the grammatical number of the VP. This latter constraint will only work if the grammar contains another rule which guarantees that the VP and the verb share their properties, for instance in a rule like

```
VP -> V NP
  VP:properties = V:properties
  NP:properties = VP:properties:object
```

When these two rules work together the parser will collect information about the subject and the object of the sentence. The nature of this information depends on the grammar writers. In general, semantic properties are included in the information structure. The semantics will at least specify that the logical or semantic subject refers to some individual with the name SADDAM, whereas the the logical or semantic object is an individual carrying the name GEORGE. These semantic entities are connected via the semantic predicate we might call DISLIKE. We use capital letters when we refer to meaning properties of the information structure in order to distinguish

the English word “dislike” from the semantic relation DISLIKE, which will receive another formal expression in a different language.

Scandinavian languages have productive compounding. Consequently, for principled reasons it is impossible to list all words in the language. New words can be created on the fly, for instance *telekommunikasjonsinteresse* (interest in telecom). Robust parsing systems must be able to recognize and semantically analyze new words of this sort, which is quite hard to accomplish safely.

4.2 Transfer

When the analysis module is finished the relevant information about the source sentence must somehow be related to information about potential sentences in the target language. In this process we might use knowledge of the language pair in question. For instance, if we translate from English into Norwegian information about subject-verb agreement in English can be omitted, since this constraint is absent in Norwegian. On the other hand, the system must collect relevant information about gender, number and definiteness for certain types of noun phrases because there is such agreement in Norwegian. Obviously, the transfer component must take care of lexical selection in the target language, that is, put together words from source sentences and target sentences. The nature of this process is often far from trivial – consider for instance *walk* and *go* in English and their Norwegian counterparts *spasere* and *gå*. *Walk* usually goes with *spasere*, but not in cases like “walk out of a meeting” with the intended meaning “to leave in protest”.

4.3 Interlingua

It is often objected that transfer strategies imply that unique transfer components must be written for every language pair, one for each translation direction, which seems to be a bad idea if there is a need for translations between many languages, as in the European Union with its many official languages. An alternative to the transfer strategy is the idea that source sentences could be translated to some general semantic form, which in turn could be used to generate sentences in many languages. This semantic form is commonly referred to as an interlingua which is assumed to have all relevant semantic properties to be found in all languages in the language “pool”. As part of a research programme for MT the interlingua idea is appealing, but as it turns out, no serious system with reasonable coverage is based on an interlingua strategy alone. The explanation is simply that a generally valid description of semantic properties common to many languages is indeed very hard to obtain. When interlingua is included in a system, it is as

a “last resort” when other strategies have failed. This low ranking stems from the annoying tendency that translation via interlingua leads to sets of translation suggestions where it is problematic to choose the best alternative, if there are feasible suggestions at all.

4.4 Generation

Generation of natural language ranges from trivial template forms to the subtleties of translation via interlingua. The key question is always “generation from what?”. A compiler in programming languages will output certain messages when the syntax of the program code is illegal, and in circumstances like this, the natural language sentences can be put more or less directly into the application program. In MT the generation phase in most cases receives some linguistic structure as a premise for sentence generation. The generator has to find sentences which are compatible with the linguistic structure given as input, although large parts of the search problem could have been resolved by some transfer module. Inevitably, semantics plays an important role for the generator. If no semantics is specified, any grammatically correct sentence will do, provided the other parts of the linguistic input structure are compatible with the grammatical properties of the generated sentence. The general rule of thumb is “the more detailed grammatical and semantic structure in the generation premise, the more precise generation results you will get”. Consequently, the preparation prior to generation becomes very important, as is the case in automatic dialogue systems; see below.

Productive compounding was mentioned in connection with parsing, and this phenomenon pops up in the generation module as well when translation goes from English to e.g. Norwegian, because a collection of English words must be glued into a novel, coherent and grammatically correct compound in the target language.

4.5 Discourse Analysis

A notorious problem in natural language analysis is how to refer to discourse referents correctly. When is it appropriate to use pronouns, when definite expressions (for instance *the solution* vs *a solution*), and when should proper names be re-entered in a text? When translation goes between European languages these problems are not so apparent because these languages share most of the relevant properties more or less directly. But if translation goes from a language where discourse referents to a greater extent are implicitly understood (like Japanese or Vietnamese) to a target language where they are grammaticalized, problems arise immediately. This is so because it is difficult to model discourse referents correctly in formal language models.

An additional problem is discourse structure and how it effects sentence structure. Basic concepts here are new and given information.

5 Relevance for Telecom Applications

After the above tour through the basics of MT we proceed to discuss some relevant aspects of MT approaches for telecom applications.

5.1 Translation with Speech Front Ends

An obvious candidate application is the idea of a translation system with speech front ends; that is, a modular system like the following:

Speech Recognition -> Analysis -> Transfer
-> Generation -> Speech Synthesis

This model was the ideal of the German *Verbmobil* project – a huge German research initiative funded by the German government from 1993 to 2000 (see <http://verbmobil.dfki.de/>). Even though *Verbmobil* did not deliver an operational system, the project has shown that grand applications of this type are conceivable, when certain important prerequisites are met, like translation within restricted domains together with general restrictions for successful use of automatic speech recognition.

5.2 Dialogue Systems

Grammars with fairly broad empirical coverage have been developed for English and German in research projects like LINGO (see <http://lingo.stanford.edu/>) and PARGRAM. The latter also includes grammar development for Norwegian, see <http://www2.parc.com/istl/groups/nltt/pargram/>. These grammars are developed for use in application types where formal grammars are needed, including MT. Dialogue systems are highly relevant in the telecom application domain because “intelligent” dialogues require both analysis (understanding what the user asks for) and generation (give answers after reasoning over properties of the application domain). Dialogue systems will enter a loop where system and user “negotiate” in order to make the system able to understand the user needs, and formal grammars can be used in these dialogues.

Ambitious dialogue applications where system control is relaxed, require that the analysis component is able to handle different ways of formulating the same message or question. Grammars with broad coverage appear to be the most salient solution, in particular if robust analysis of unknown words is possible. Some version of a finite state transducer is the standard technique used for analysis of unknown words.

5.3 Limitations of Rule-based Approaches in the Speech Domain

Modern speech technology is dependent on stochastic methods, simply because these methods give the best results, both with respect to real time performance and accuracy. Rule-based systems in language technology tend to be unable to scale up when exposed to realistic application domains outside the labs where good real time performance is required (finite state methods escape this generalisation). If sophisticated rules systems are being built, then powerful description languages will be used, and, consequently, the worst case complexity properties of the systems are not appealing. This is not to say that rule-based techniques are irrelevant “in the real world”, but one should be careful if rule-based methods are imported into e.g. automatic telephone applications with speech front ends. Applications where real time requirements could be relaxed, as in the SMS domain, rule-based systems are however highly relevant, see for instance Amble 2000 (and the SMSTUC application www.lingit.no).

6 Further Reading

The present article provides a very brief sketch of basic problems in MT. Readers who would like to explore this field in more detail should consult Dorr, Jordan & Benoit (1999), which is an excellent introduction and overview of the field with all relevant references given. The standard reference for unification is Shieber (1986), but Jurawsky & Martin (2000) also provide a good exposition. In addition, this book offers an overview of language technology in general, and it is widely used as textbook at universities around the world. A thorough introduction to formal semantics and discourse analysis is Kamp & Reyle (1994).

Literature

Amble, T. BusTUC – A Natural language bus route oracle. *Proceedings of the ANLP-NAACL 2000 Conference*. Morgan Kaufmann Publishers, 1–6, Seattle, WA, USA, 2000.

Dorr, B J, Jordan, P W, Benoit, J W. A Survey of Current Research in Machine Translation. Zelkowitz, M (ed). *Advances in Computers*, 49, 1– 68, 1999. London, Academic Press.

Jurawsky, D, Martin, J H. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Upper Saddle River, NJ, Prentice Hall, 2000.

Kamp, H, Reyle, U. *From Discourse to Logic : Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht, Kluwer, 1993.

Shieber, S. An Introduction to Unification-Based Approaches to Grammar. *CSLI Lecture Notes*, 4, 1986.

Speech Recognition – a Tutorial and a Commentary

FRANK K. SOONG AND BIING-HWANG JUANG



Dr. Frank K. Soong has worked on a wide variety of topics, including speech coding, speech recognition, speaker recognition, and room acoustics. His recent interests in hands-free speech recognition helped the development of a voice interface for mobile applications. He was co-recipient of the Lucent President Golden Award for developing the Bell Labs Automatic Speech Recognition (BLASR) software package. He has served the IEEE Speech Technical Committee as committee member and as associate editor of the *Transactions on Speech and Audio*. After 20 years of research at Bell Labs, he retired as a Distinguished Member of Technical Staff in 2001 and is currently an invited researcher at the Spoken Language Translation Labs, ATR, Japan. frank.soong@atr.co.jp



Dr. Biing-Hwang Juang has been engaged in a wide range of research activities, from speech coding, speech recognition, and intelligent systems to multimedia and broadband communications. He has published extensively and holds a number of patents. He received the Signal Processing Society's Technical Achievement Award in 1998 and was named the Society's 1999 Distinguished Lecturer. In 2000, he was awarded the IEEE Third Millennium Medal for his contributions to the field of speech processing and communications. He became a Bell Labs Fellow, the highest honor in technical fields at Bell Labs, in 2000. He is a Fellow of the IEEE. Dr. Biing-Hwang Juang is currently a chair professor at the Georgia Institute of Technology.

This article reviews the key components in the modern day speech recognizers and related advances. Topics covered include fundamental concepts and building blocks of a recognizer, speech recognition task categorization, historical perspective of technology development, state of the art and future challenges.

1 Introduction

Speech generation is a transduction process in which the talker's essential idea (e.g. to invite someone to lunch) is being transduced to a sequence of words (e.g. "Aren't you hungry? Would you like to go to lunch together?"), which are then realized by human's articulatory apparatus in the form of acoustic signal, propagating over the air to reach a listener. Speech recognition, in a strict sense, is the reverse of this process, aiming at recovering the words that are embedded in the acoustic signal, and the original notion, carried by a sequence of words. Automatic speech recognition is the task of a machine, most likely a computer, which carries out this reverse process. A speech signal (the so-called acoustic events) may encompass linguistically relevant components, such as words or phrases, as well as irrelevant components, such as ambient noise, and extraneous or partial words. The existence of irrelevant components (noise and extraneous sounds), distortions (undesirable modification of the sound quality), and the ambiguity in the transduction process (e.g. imprecise articulation or mispronunciation) is what makes the task of automatic speech recognition challenging.

Speech recognition may sometimes include speech understanding in a casual sense. This is because speech recognizers are often designed to take in natural spoken sentences and used for spawning certain actions in a limited domain, such as routing a call to an extension (e.g. "Please connect me to Fred, the manager" given to an automatic call router). In performing this kind of tasks, rather than converting sounds into words in a one-to-one manner, the recognizer is usually constructed to focus on the talker's notion (or some keywords in the sentence, "Connect", "Frank" in the example) from the speech signal and ignore the rest of the signal. Here, the recognizer does not really "understand" speech but only appears to, because the total ideas carried in a sentence for such an application are limited and the finite possibilities can be uncovered by mapping "spotted" keywords to a prescribed set of actions. This technology is generally referred to as "keyword spotting", in contrast to the above sound-to-word

conversion process which is referred to as "transcription".

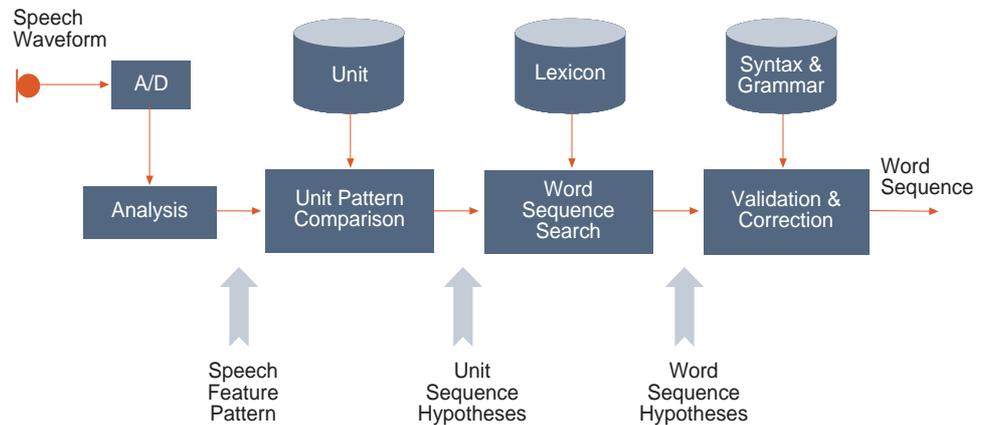
In this paper, we will present the fundamental concept and techniques used in an automatic speech recognition system. We will explain state-of-the-art technologies and their performances in the form of benchmark for future references. We conclude our presentation with a discussion on remaining challenges in the field of automatic speech recognition and its extension to spoken language understanding.

Before we proceed, it should be understood that while the field is still under active research, existing technologies are mature enough to support a wide range of applications such as an automatic voice activated typewriter that responds to voice in a quiet room, voice-controlled devices or access to information services (e.g. news and stock quotes), and automated commercial transactions (e.g. account balance inquiry or merchandise order by telephone), just to name a few.

2 Fundamental Concept and Building Blocks of a Recognizer

Figure 1 illustrates the essential steps in converting speech signals into a sequence of words. First, a microphone picks up the sound in the ambient in which the speech signal is being generated and turns it into an electrical signal. A device called Analog-to-Digital (A/D) converter is used to transform the analog electrical signal into a digital representation, ready to be processed by a digital processor or computer. The digital computer performs a set of algorithms, or computing procedures, to accomplish the sound-to-word process, which begins with "feature analysis". The purpose of feature analysis is to extract the essential and salient characteristics, which are critical to the automatic recognition process, of the input speech signal. The result is a set of so-called features or feature vectors. The most prevalent feature of speech is one that is derived from its *short-time spectrum*, which is obtained by applying short-time spectral analysis over a window of the speech data, usually 20–30 ms in duration. Successive application (say

Figure 1 Basic components of a speech recognition system



every 10 ms) of short-time spectral analysis over the entire duration of the signal produces a sequence of spectral representations, which forms a vectorized speech feature pattern, in a two-dimensional, time and frequency domain.

A store of phoneme patterns or models need to be established beforehand through a training procedure, called “acoustic modeling”, which finds the elements of speech, i.e., essential characteristics of the acoustic realization of the “phonemes”, and organizes them to represent the fundamental units of the language when realized in spoken form. The speech feature pattern is compared to the units in the store to generate a unit sequence hypothesis or a set of such conjectured hypotheses. A speech signal has substantial variations along many dimensions. First and probably the most obvious one is the speaking rate. A human speaker cannot normally reproduce at wish an utterance of identical duration. Second, variation in articulation is inevitable both in terms of talker-specific characteristics (e.g. differences in vocal tract length and shape) and in the manner a phoneme is produced. Third, the pronunciation variation occurs across speakers (e.g. dialect or accent) as well as speaking contexts (e.g. some phonemes may be dropped in casual conversation). A procedure called *Dynamic Programming* is used during unit comparison to generate the best match by normalizing, i.e., stretching or compressing (i.e. time normalizing), the temporal pattern (so-called “*Dynamic Time Warping*”) and by conjecturing how a phoneme may have been produced in a statistical sense, to account for the aforementioned variations. The purpose of training is to obtain knowledge of such statistics. Training is performed over a large collection of data, with known labels or identity, according to a procedure and the statistical knowledge is encapsulated in a mathematical formalism, the most widely adopted one of which is the so-called “hidden Markov model (HMM)”.

A hidden Markov model is a doubly stochastic process involving (at least) two levels of uncertainty. It is a finite state (say N-state) machine in which each state is associated with a random process. The random process in each state accounts for the variation in the realization of a speech unit (such as a phoneme), across time as well as across speakers and speaking contexts. The finite state machine is a Markov chain, which characterizes the probabilistic relationship among the states in terms of how likely one state may follow another. Depending on the designation of the state, this inter-state relationship can be representative of the nature of phonology and syntax of the language. Figure 2 illustrates a 3-state hidden Markov model.

The resultant hypothesized unit sequence is then matched to the stored dictionary, or more specifically the lexicon, which is the collection of target words each being represented by a unit sequence, to reach a tentative decision on the word identity. The decoded word sequence is further subject to verification and validation, to ensure its conformance to syntactic and grammar constraints. Often, these constraints are introduced even during the word sequence search stage to eliminate unnecessary errors early on to prevent the so-called error propagation and to increase the search efficiency (meaning to reduce the possible hypotheses in the search space).

Syntactic and grammar constraints can be expressed as deterministic rules or as statistical models. The former defines permissible word sequence relationship deterministically and the latter associates probabilities with these relationships. In general, they are referred to as the “language model”, in relation to the aforementioned “acoustic model”. The most frequently used language model is based on N-gram statistics which assigns a probability to the occurrence of a present word, conditioned on the N-1 words that appeared before it. The finite state nature of the language model renders itself synergistic with

the hidden Markov model, used for acoustic modeling, and amenable for efficient computation in the search.

Training of a statistical language model concerns with the procedure to obtain the proper statistics in terms of word sequence relationship (which words and how often they are likely to form a sequence one next to the others). To ensure the probability estimate is accurate, the procedure must involve a large database of text, which should be sufficiently rich in various kinds of sentential expressions to reflect the language. Insufficient data for training will lead to either unreliable probability estimate or poor “coverage” – meaning, many expressions the recognizer experiences may not have appeared, or not sufficiently in the database for training and thus have un- or ill-defined probability.

3 Speech Recognition Task Categorization

Machine recognition of speech has neither followed nor achieved what a human listener is generally capable of doing. In fact, the development of automatic speech recognition technologies in the past often assumes a certain categorization of the speech recognition task due to this lack of generality in the system capability. For example, a system that is designed for a specific talker, the so-called speaker-dependent system, would have a much simpler architecture than a “speaker-independent” system that is meant to deal with users in the general population.

Similarly, in terms of acoustic realization of linguistic expressions, a sequence of words may be uttered individually in isolation or in a connected manner without clear, intentionally controlled articulation gaps that separate the words. A system designed for the former is called an “isolated (or discrete) word recognizer”, while for the latter a “connected word recognizer” (e.g. dealing with a digit sequence spoken without intentional pauses) or a “continuous speech recognizer” (e.g., dealing with naturally spoken paragraphs of an article).

In the case of continuous speech recognition, one needs to be cautious about the implicit spontaneity in the utterances. The speaker’s articulations can be drastically different between reading a prepared text, such as a well-written newspaper article, and conversing freely and impromptu with another person. In the former case, words and phrases are normally well articulated with few “disfluencies” and sentences follow syntactical and grammatical rules with few exceptions. Recognition of such kind of utterances is relatively much easier than that of a conversational speech which can be significantly ill-formed and with a substantial amount of dis-

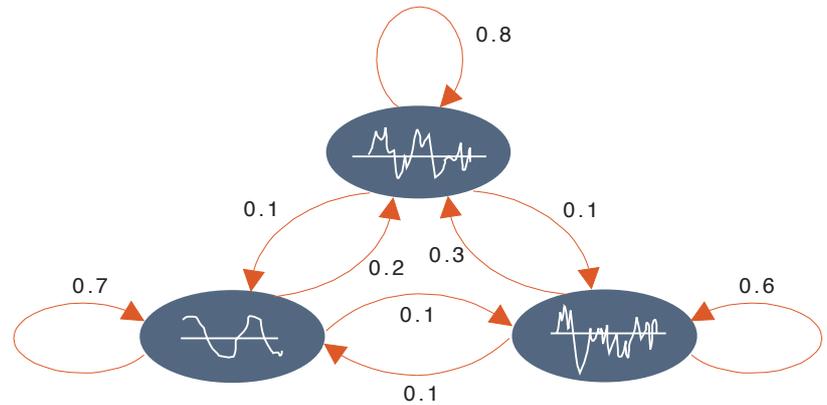


Figure 2 A 3-state hidden Markov model

fluencies, such as partial words, stuttering, repairs, etc. or incomplete sentences. When disfluencies occur, the pre-established decoding scheme would usually fail because, so far, modeling of disfluency proves to be difficult and has not been cogently formulated and built in any practical engineering systems to perform real tasks.

Another dimension in speech task categorization is the degree of sophistication, or complexity, of the task. The complexity of a task is a strong function of the size of the vocabulary that the system is designed to deal with, and the perplexity of the grammar and syntactic constraints. A task involving a few tens of words in the vocabulary is generally called “small vocabulary recognition,” and is mostly used for command-and-control applications with isolated word utterances as input to the recognizer. In these command-and-control applications, there are usually very few grammatical constraints. When the vocabulary grows to a few hundred, it becomes possible to construct meaningful natural sentences, although the associated grammatical rules are still fairly rigid (e.g. the Naval Resource Management task in the DARPA research program). For dictation, report writing, or newspaper transcription, a large vocabulary (may be well beyond a thousand words) system is generally needed.

A large vocabulary continuous speech recognition system has a set of acoustic models that probabilistically characterize the (almost) entire acoustic space associated with speech, and a language model with a sufficient coverage of linguistic expressions for the language used in the task. Depending the task, the set of linguistic expressions in the task may be rather limited, or may be uncountably large. The complexity is often measured in terms of “perplexity” which can be considered as the average word branching factor, meaning the average number of distinct words that can follow a word. The higher the perplexity, obviously, the harder the recognition task as the recognizer needs to deal with a higher

degree of uncertainty. A “language” generated with a stylized grammar such as that in the Resource Management task usually has a perplexity in a few tens. Newspaper articles have a nominal perplexity on the order of a hundred. Therefore, when describing a continuous speech recognition task, it is informative to include an estimate of its perplexity, qualified by the language model used in the estimation process.

4 Technology Development – A Historical Perspective

Speech recognition technology today is the result of decades of research by a large number of scientists and engineers, and the financial support from many governmental as well as industrial organizations. Figure 3 compiles a number of major steps that contributed to the advance in speech recognition, providing a historical perspective for the development of such an important technology.

Automatic speech recognition before the 1950s was viewed mostly from the acoustic-phonetic and linguistic perspectives. Many acoustic-phonetic studies aimed to discover the key “features” of the acoustic event that are associated with a phoneme, which is commonly viewed as the most basic unit of speech. These features were discussed both from the angle of speech articulation, such as place and manner of articulations, and the angle of speech (auditory) perception, such as the just noticeable difference (JND) or difference limen (DL) (e.g. in terms of formant frequency, bandwidth and amplitude) and the articulation index. The speech spectrogram, the so-called “visible speech”, developed in the 1940–50s by displaying the time-varying

energy distribution of the speech signal in different frequency bands has made the study of acoustic-phonetics more ready for engineering uses. These studies help researchers tremendously in the understanding of the speech signal, but have not yielded directly a practical and successful method for automatic speech recognition.

Starting from the late 1950s, engineers started to build and experimented with systems that are designed for only a very small set of words. While they followed the suggestions from acoustic-phonetic studies, the idea of pattern matching (inferring the identity of the unknown pattern by matching it to pre-stored templates with known identities) emerged as the basic architecture of practical system designs. The focus of inquisition was then shifted from discovery of acoustic-phonetic features to distance measures (which measure the similarity between two patterns in a intuitively appealing and objectively tractable way) and the algorithms for temporal alignment (to deal with the very fundamental time variation in speech production as mentioned above). The success in the cepstral distance measure and the rather thorough study of the dynamic programming based dynamic time warping algorithm came about in the late 70s and early 80s. Successes in these areas also reinforced the concept and applicability of the formal statistical pattern recognition methodology in automatic speech recognition.

An important basis of the statistical pattern recognition approach is Bayes’ decision theory, which formulates the pattern recognition problem as a problem of statistical inference using knowledge of the joint distribution of the ob-

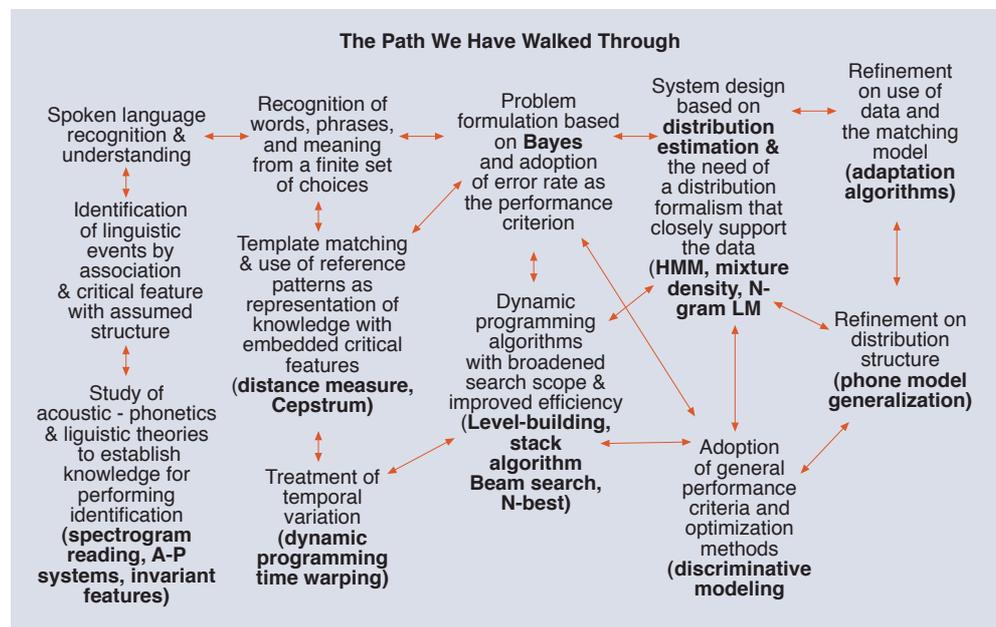


Figure 3 A historical perspective of the development of speech recognition techniques

served pattern and its identity. This provides a tractable methodology for the design of a pattern recognition system. When the bridge between pattern matching and probability (or likelihood) assessment was established for the speech recognition problem in the early 80's, the statistical method became a preferred choice due to its algorithmic tractability and engineering pragmatism.

Another major advance came about when hidden Markov model was proposed, first in the late 60's for sequence analysis, and then mid-70s for speech modeling. The formalism of hidden Markov model works well with the concept of dynamic programming, which is a critical component in pattern sequence matching, and is readily useful as a means to approximate the probability distribution of speech. Today, most if not all speech recognizers are based on hidden Markov models.

As the architecture of hidden Markov model is applied to large vocabulary continuous speech recognition, it was found that elaborate model structures that take into account the dependency of the speech signal upon neighboring phonemes (or events) are helpful and necessary. The practice of context-dependent statistical modeling of acoustic events thus became widespread. It is now a common scheme that a large vocabulary speech recognition system uses thousands of context-dependent subword models, rather than just a few tens according to the list of phonemes of the language.

Recent improvements to the fundamental methods of statistical pattern recognition have focused on the issues of distribution mismatch (between the observed data and the chosen form of the observation distribution), which has to be circumvented in order to achieve a potential optimal recognition accuracy, and of condition mismatch (between the operating and the training conditions, say noisy and quiet conditions) which causes degradation to the recognizer's performance. New concepts in discriminative modeling, which aim at direct minimization of the recognition error by introducing a proper optimization criterion to the training process, have led to substantially high recognition accuracy in real world applications and tasks. Just as active is the area of adaptation, or adaptive modeling, which tries to find a proper way to mitigate the problem of condition mismatch by either rapidly adapting the pre-stored templates or statistical models to the field data as the system is being used, or transforming the observed data such that the model likelihood computation will produce reliable results.

5 State of the Art

Since the early 1990s, we have started to witness modest success of the automatic speech recognition technology in limited applications. These limited successes usually appeared in the area of small to medium size vocabulary recognition for well-defined tasks, such as interactive voice response systems with voice control (e.g. the Conversant System of AT&T and later of Lucent), credit account inquiries (e.g. the AT&T Universal Card Services that automate millions of calls every month), dictating an X-ray readout by a radiologist, voice dialing of telephone numbers, or voice-enabled call routing with a private bureau exchange (PBX) system. Towards the late 1990s, further advances made possible tasks like voice-activated word processing and document editing, voice-browser for Internet surfing, and voice-directed advanced information services such as stock quote and news headlines over the telephone.

Each of these applications demands a certain degree of recognition accuracy for the deployment of the service to be viable. While these performance figures are task dependent and often kept only for the deploying company's internal reference, we review typical speech recognition performances for some benchmark tasks here to provide a general perspective for the maturity of the technology. Table I shows the average word error rate for isolate word recognition without grammar constraints achieved by both speaker-dependent (SD) and speaker-independent (SI) systems. Generally speaking, the performance is more sensitive to the confusability in the vocabulary than to the size of the vocabulary. The 39-word alpha-digits vocabulary (consisting of letters A through Z, digits 0 through 9, and three additional command words) contains several highly confusable word subsets, such as the E-set (B, C, D, E, G, P, T, V, Z and digit 3) and the digit three, and the A-set (A, J, K and digit 8). A state-of-the-art SI system can only achieve a word accuracy of about 93 %. In comparison, a

Table I Performance of isolated word recognition systems in various tasks

| Recognition Task | Vocabulary Size | Mode | Word Accuracy |
|------------------------------|-----------------|------|---------------|
| Digit (0-9) | 10 | SI | ~100 % |
| Voice Dialing | 37 | SD | 100 % |
| Alpha-digits & Command Words | 39 | SD | 96 % |
| | | SI | 93 % |
| Computer Commands | 54 | SI | 96 % |
| Air Travel Words | 129 | SD | 99 % |
| | | SI | 97 % |
| City Names (Japanese) | 200 | SD | 97 % |
| Basic English Words | 1109 | SD | 96 % |

| Recognition Task | Vocabulary Size | Perplexity | Word Accuracy |
|----------------------------------|-----------------|------------|---------------|
| Connected Digit Strings | 10 | 10 | ~99 % |
| Naval Resource Management | 991 | <60 | 97 % |
| Air Travel Information System | 1,800 | <25 | 97 % |
| Business Newspaper Transcription | 64,000 | <140 | 94 % |
| Broadcast News Transcription | 64,000 | <140 | 86 % |

Table II Performance benchmark of various continuous speech recognition tasks

task involving 1,109 common English words (in isolation) can be accomplished with a word accuracy of 96 %.

For connected-word and continuous speech recognition, as mentioned above, it is important to associate a task with an estimate of the perplexity of the task. Table II provides a performance benchmark for various continuous speech recognition tasks.

As mentioned above, the Naval Resource Management task involves a language used in military duties and highly stylized with a perplexity of no more than 60. In the Air Travel Information System (ATIS), the utterances are usually queries of flight information such as airfare, flight time, the origin and destination cities, and so on, and are limited in structural variations, even though the sentences appear to be natural. It has a perplexity of less than 25. The Business Newspaper Transcription task involves “read” speech – the speakers read to a fixed mounted microphone the text of news articles in the Wall Street Journal newspaper. Utterances in the Broadcast News Transcription task, nevertheless, are the so-called “found” speech, such as news announcement from radio or television stations, and may be mixed with background noise or music, or distorted during radio transmission. Obviously, such task would be more difficult than transcribing a news article in a quiet room with a well-mounted microphone. The recognition accuracy reflects well the implicit complexity or difficulty of the task.

6 Further Challenges

The rapid advancement of the automatic speech recognition technology has pushed the frontiers of human machine communications into a brand new era. Newer and more ambitious applications have been field tested and introduced as services or products, like the AT&T’s “How May I Help You?” System for routing a customer’s request to an appropriate department, IBM’s “Via Voice” speech-to-text transcription system for preparing a first draft of a letter or memo, United Airlines flight information system to answer questions regarding to arrival, departure and other information regarding to United Airlines flight schedule, to name just a few most noticeable ones.

Again, we should not be too excited by the success or enthusiastic receptiveness of these new applications. Human-machine communications based upon the current technology are still in their infancy and the success of any new applications, if any, is by all means limited. When compared to human’s speech recognition, understanding and responding capabilities, the current ASR technology based systems still have a long way to go.

Normal human beings have a much sharper hearing than what the current best ASR systems have, especially in a noisy environment. Humans, in general, can also make a much better sense of ill-formed input speech than the best machine can. Stuttered, ungrammatical, spontaneous and conversational speech does not seem to cause any hardship to a human listener. Sometimes those ill-formed input speech are not even noticed by the listener. But the current speech recognizers experience tremendous difficulties in dealing with such kind of corrupted input and as a result, the recognition performance degrades dramatically down to a level which renders the ASR system not usable.

To face these new challenges, new knowledge or quantifiable knowledge that can be modeled and built into the next generation ASR is absolutely necessary. The questions are what is the knowledge and how to acquire and to quantitatively characterize it. We feel that to understand how humans do it is crucial. Even though by mimicking every details of bird many not be adequate, even harmful for us to learn how to fly or build something that can fly. But to learn how a bird can fly gracefully and sometimes even effortlessly, is always helpful and insightful to us. Similarly, to understand how human beings can separate speech from background noise or interfering speech, or to bring our attentions to such fascinating phenomena of human hearing mechanism is useful. Speech researchers can thus learn how to develop new robust algorithms for the next generation ASRs.

Efforts in these directions can roughly be put into the following categories:

- 1 Robust auditory peripheral including cochlear and hair cell modeling in human inner ear; masking effects in the auditory peripheral and neuron firing rate, etc.
- 2 Auditory scene analysis at the cortex where different auditory scenes can be grouped into relevant clusters and different but overlapped in time and frequency scenes can be separated. Purely information theory based approach like independent component analysis (ICA) has been shown effective in separating indepen-

dent sound sources blindly using the infomax principle with certain mild constraints.

- 3 Discrimination of one sound from the other in human auditory cortex, including: the magnet theory of vowel prototypes in human auditory cortex and experiments of training Japanese native speakers to differentiate the two semi-vowels, “l” from “r”, which are indistinguishable in Japanese. Researchers of the artificial neural networks have advocated the ideas to train highly discriminative neural nets to simulate this specific behavior. More cogent statistical approach have also been adopted by speech researchers to train highly discriminative HMM models using the Minimum Classification Error (MCE) or the Maximum Mutual Information (MMI) principles.
- 4 Missing feature theory in human perception where human beings have been found capable of filling the missing (or occluded) acoustic cues due to mis-production or noise masking. It is also reported that human beings can adaptively weight noisy acoustic features in terms of their relative contaminations (by interferences) such that their hearing in a noisy environment can degrade more gracefully. In the HMM context, marginalization (ignoring or weighting down uncertain features) and data imputation (estimating noise contaminated parameters using an optimal criterion like the Maximum A Posterior (MAP) to obtain a conditional mean) have shown reasonable success in improving recognition performance in noise.
- 5 Human listeners do not have any serious problems, or they can adapt themselves easily, to recognize speech input from different speakers of different gender, age, accents, different environments or transmission media, etc. HMM adaptation has been the central research focus to achieve similar goals when the training and test speech are highly mismatched with each other.

The speech recognition research, an interdisciplinary field, requires a very wide range of knowledge from both speech production and speech perception. In order to make ASR more robust to interference, more discriminative to differentiate similar sounds and more versatile to handle large vocabulary, continuous, speaker independent, spontaneous speech input, we need to tackle the problem from various angles and to gain some true understanding of the whole speech chain, from production to the perception. In the mean time, before we obtain a universal solution to the general ASR problem, certain part of the technologies are matured enough to successfully support some useful, hopefully lucrative, applications. In search of the so-called

“the killer application”, we need first to understand the advantages and limitation of the state-of-the-art ASR technology, then to sift through an endless list of possible application scenarios before we can find the best marriage between the current technology and a matched applications.

7 Suggested Readings

Allen, J B. How do humans process and recognize speech? *IEEE Trans Speech and Audio Proc.*, 2 (4), 567–577, 1994.

Baum, L E, Petrie, T. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.*, 37, 1554–63, 1995.

Bell, A J, Sejnowski, T J. An information-maximization approach to blind separation to blind separation and blind deconvolution. *Neural Computing*, 7 (6), 1004–1034, 1995.

Bregman, S. *Auditory Scene Analysis*. Cambridge, MA, MIT Press, 1990.

Cooke, M P, Green, P D. *Modelling Auditory Processing and Organization*. Cambridge, UK, Cambridge University Press, 1993.

Duda, R, Hart, P. *Pattern Classification and Scene Analysis*. Wiley, NY, 1973.

Katagiri, S (Ed.). *Handbook of Neural Networks for Speech Processing*. Norwood, MA, Artec House Inc., 2000.

Lee, C-H, Soong, F K, Paliwal, K K (Eds.). *Automatic Speech and Speaker Recognition: Advanced Topics*. Boston, MA, Kluwer Academic Publishers, 1996.

Lee, K F. *Automatic Speech Recognition – The Development of the SPHINX System*. Boston, MA, Kluwer Academic Publishers, 1989.

Rabiner, L R, Juang, B-H. *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, Prentice Hall, 1993.

Rabiner, L R. A Tutorial on hidden Markov model and selected applications in speech recognition. *Proc. IEEE*, 77 (2), 257–286, 1989.

An Overview of Text-to-Speech Synthesis

PER OLAV HEGGTVEIT



Per Olav Heggveit (37) is a research scientist at Telenor R&D. He graduated from the Norwegian University of Science and Technology as *Siv.Ing. (MSc)* in 1989. In 1990–1991 he worked as product manager for speech products at Falck Produkter AS. While finishing part-time studies in computational linguistics at the University of Oslo, he started in 1991 as a research scientist at Telenor R&D working mainly with text-to-speech synthesis, and in particular with text analysis and prosody modelling. Heggveit's recent work includes IP-based voice services, unit selection synthesis and voice enabled web services.

per-olav.heggveit@telenor.com

This paper will provide a brief overview of text-to-speech (TTS) synthesis. TTS is defined here as computer generated speech used to translate electronically stored text information into aural information. The topics covered include TTS applications, milestones in TTS history, components of a typical TTS system, signal generation methods, state-of-the-art and current research trends, standard application frameworks such as APIs and Markup Languages, and commercial systems.

Text-To-Speech (TTS) synthesis can be defined as computer-generated speech used to translate electronically stored text information into aural information. The more general term *speech synthesis* is used when also talking about synthesis independently of the input format, such as simple playback of stored words, phrases and sentences and also about the generation of speech from any other input source than text. Speech synthesis has a long history, and even though TTS has been around for many decades, research and development in this field are more extensive than ever.

Applications – What is TTS Used For?

Speech is the most natural way of human-to-human communication. Synthetic speech is a method of copying the generative part of natural speech communication. Speech synthesis can help humans and machines to communicate in a more natural way. The range of applications is wide, from simple talking clocks, calculators and telephone messaging systems with a small and limited vocabulary, to 3D talking heads with synthesis of unrestricted text and synchronisation of head and mouth movements.

Applications for the Handicapped

The blind have been pioneers in applying TTS in everyday communication. Their strong need to get text read aloud has made them generally more willing to accept the low quality of avail-

able TTS systems in the past, and they became very early adaptors to this technology. Audio books existed long before commercially available TTS systems, but the production of audio books is expensive and time consuming. A lot of text such as newspapers, magazines and mail is never read and recorded as audio books. TTS introduces freedom and flexibility to read any text any time. When Raymond Kurzweil in 1976 introduced the *Kurzweil Reading Machine* (Kurzweil 1990) with an optical scanner, character recognition and TTS, a whole new world of paper text reading became available to the blind.

From its origin as a special purpose reading machine, this technology has moved on to general purpose PCs. With a scanner, software OCR and TTS, it is easy and affordable to make a normal PC into a reading machine. Today, TTS is a very useful tool for blind PC users reading e-mail, news, and other web content.

Synthesized speech also gives the deaf and vocally impaired a possibility to communicate with people who do not understand the sign language. The well-known astro-physician Stephen Hawking gives his lectures using TTS.

The use of TTS can be more difficult and frustrating for vocally impaired than for blind users. The TTS voice is a replacement of their personal voice. It should therefore be as representative as possible for each individual in age and dialect, and it should also be able to convey emotions such as happiness and sadness. Most available TTS systems have only a relatively small number of different voice personalities and the ability to convey emotions is more or less absent.

Education

TTS can be used in many educational situations. A computer with a speech synthesizer is an always available and patient teacher. It can be used in teaching native or foreign languages. The TTS system can spell and pronounce syllables, words, phrases and sentences helping the student as needed. It can also be used with interactive educational applications.



Stevie Wonder with his *Kurzweil Reading Machine* (Kurzweil 1990)

Especially to people who have reading impairments (dyslexics), speech synthesis may be very helpful.

TTS in connection to a word processor is also a helpful aid to proof reading. Many users find it easier to detect grammatical and stylistic problems when listening than when reading. Normal misspellings are also easier to detect.

Multimedia and Telecommunications

TTS has been used for many years in different kinds of telephone enquiry systems. The text information presented as speech ranges from simple program announcement of cinemas, theatres, exhibitions, etc., to presentation of results from huge database information queries.

In *Unified Messaging* systems and *talking e-mail readers*, you get e-mail read aloud over the telephone line. The email headers and content is presented to the user as synthesized speech. *SMS-to-Voice* services are also available where you get the SMS messages read aloud instead of reading the messages yourself. The most successful application of talking e-mail seems to be as an integrated part of unified messaging systems as a complementary service to other ways of accessing your messages. Most of the main messaging platform vendors include such functionality.

Automatic Reverse Directory services are offered in several countries with commercial success (Nebbia et al. 1998). In this service you input a telephone number by voice or keypad and you get the name and address read back to you. Most of the text synthesized is name and address information that is quite different from general text. The sentence format is relatively simple and more or less the same for each utterance, but the pronunciation can vary a lot depending on the origin of the name.

TTS also plays an important role in *virtual personal assistants* that route calls, take messages, manage contacts, dial outgoing calls, etc. Automatic speech recognition (ASR) is used to control the assistant and TTS to get textual information back from the assistant. A well-known commercially available voice-activated assistant is Wildfire (www.wildfire.com) owned by Orange.

Another range of voice applications is called *Voice Portals* providing callers with access to news, stock quotes, movie listings, driving directions, and more.

Auto attendants allow a customer to say the name of the department or individual they wish to contact using ASR to recognize the names and TTS to read the recognized names back to the caller for confirmation.

The newest applications in speech synthesis are in the area of multimedia and *multi-modal speech synthesis*. A popular and well-known example is the web based Ananova Video reports service. A synthetic animated female is reading the news stories much like a real live news reporter would do. The difference is that Ananova reads for you anytime (www.ananova.com). Visual head and face synthesis can also make TTS easier to understand and more natural, just as it is easier to understand a human when you see his or her face.

Another use of TTS is in speech-to-speech *language translation systems*. Such systems recognize spoken input, analyze and translate it, and finally utter the translation in the language translated into. In the project VERBMOBIL three business-oriented domains, with context-sensitive translation between three languages (German, English, and Japanese) is implemented (Wahlster, 2000).

History – Milestones in TTS History

In 1791 Wolfgang von Kempelen introduced his “Acoustic-Mechanical Speech Machine”, which was able to produce single sounds and some sound combinations (Klatt 1987, Schroeder 1993). Later, in the mid 1800s Charles Wheatstone constructed a more complicated version of von Kempelen’s speaking machine, which is shown in Figure 1. It was capable of producing vowels and most of the consonant sounds, as well as some sound combinations and full words. Vowels were produced with a vibrating reed. Deforming the leather resonator made voice resonances. Consonants were produced with turbulent flow through a suitable passage with no reed vibration.

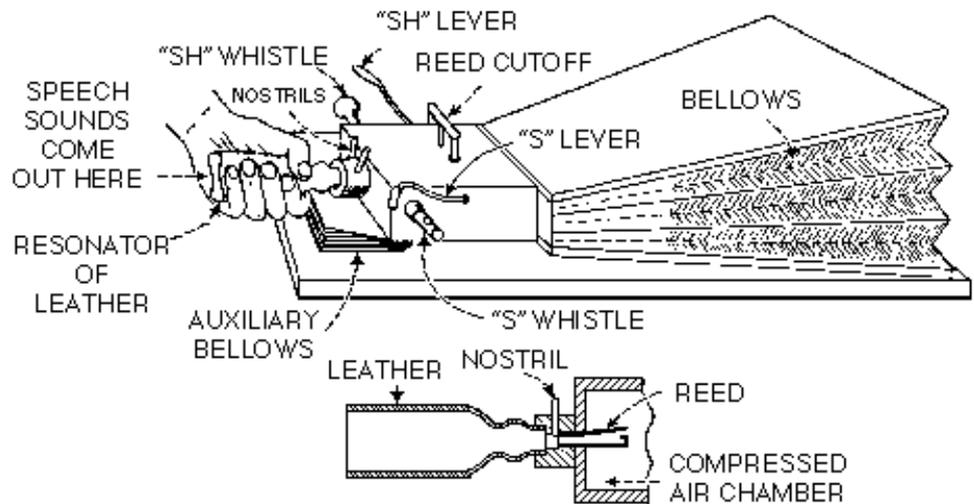
The young Alexander Graham Bell, inspired by Wheatstone’s speaking machine, in the late 1800s constructed his own complex speaking machine, which was able to produce a few simple utterances. Bell also made some experiments with his terrier. He put the dog between his legs and made it growl continuously, and then manipulated the vocal tract by hand to produce speech-like sounds (Flanagan 1972, Schroeder 1993). After a long period of training he made his dog utter the sentence “How are you, Grand-mamma?”.

Stewart introduced the first electrical synthesis device in 1922 (Klatt 1987). The synthesizer had a buzzer as excitation and two resonant circuits to model the acoustic resonances of the vocal tract (formants). The machine was able to generate single static vowel sounds with two lowest formants, but not any consonants or connected utterances.



Ananova, an artificial news reporter (www.ananova.com)

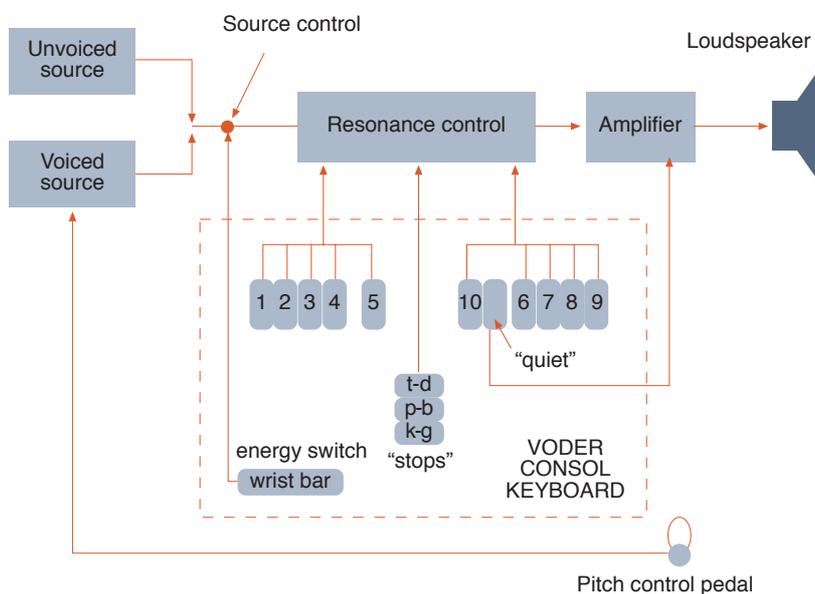
Figure 1 Wheatstone's reconstruction of von Kempelen's speaking machine (Flanagan 1972)



The first device to be considered as a speech synthesizer was VODER (Voice Operating Demonstrator) introduced by Homer Dudley in New York World's Fair 1939 (Flanagan 1972, Klatt 1987). The VODER consisted of a wrist bar for selecting a voicing or noise source and a foot pedal to control the fundamental frequency. The source signal was routed through ten band pass filters whose output levels were controlled by fingers. It took considerable skill to play a sentence on the device. The speech quality and intelligibility were far from good but the potential for producing artificial speech was well demonstrated.

After demonstration of VODER the scientific world became more and more interested in speech synthesis. It was finally shown that intelligible speech could be produced artificially. The basic structure of VODER is very similar to present systems based on the source-filter-model of speech.

Figure 2 The VODER speech synthesizer (Klatt 1987)



The first formant synthesizer, PAT (Parametric Artificial Talker), was introduced by Walter Lawrence in 1953 (Klatt 1987). PAT consisted of three electronic formant resonators connected in parallel. The input signal was either a buzz or noise. A moving glass slide was used to convert painted patterns into six time functions to control the three formant frequencies, voicing amplitude, fundamental frequency, and noise amplitude.

At about the same time Gunnar Fant introduced the first cascade formant synthesizer OVE I (Orator Verbis Electricis) which consisted of formant resonators connected in cascade. Ten years later, in 1962, Fant and Martony introduced an improved OVE II synthesizer, which consisted of separate parts to model the transfer function of the vocal tract for vowels, nasals, and obstruent consonants (Fant and Martony 1962). Possible excitations were voicing, aspiration noise, and frication noise. The OVE projects were followed by OVE III and GLOVE at the Kungliga Tekniska Högskolan (KTH), Sweden, and the present commercial Infovox system is originally descended from these (Carlson et al. 1990).

In 1958, George Rosen introduced the first articulatory synthesizer, the DAVO (Dynamic Analog of the VOcal tract) (Klatt 1987). The synthesizer was controlled by tape recording of control signals created by hand.

In the mid 1960s, the first experiments with Linear Predictive Coding (LPC) were made (Schroeder 1993). Linear prediction was first used in low-cost systems, such as TI Speak'n'Spell in 1980, and its quality was quite poor compared to present systems, but with some modifications the method has been found useful in many present systems.

The first full text-to-speech system for English was developed in the Electrotechnical Laboratory, Japan, 1968 by Noriko Umeda and his companions (Klatt 1987). It was based on an articulatory model and included a syntactic analysis module with sophisticated heuristics. The speech was quite intelligible but monotonous and a far cry from the quality of present systems.

In 1979 Allen, Hunnicutt and Klatt demonstrated the MITalk laboratory text-to-speech system developed at M.I.T. Two years later Dennis Klatt introduced his famous Klattalk system, which used a new and sophisticated voicing source (Klatt 1987). The technology used in MITalk and Klattalk systems still forms the basis for some of the present commercially available synthesis systems, such as DECTalk.

In the late 1970s and early 1980s, a considerable amount of commercial text-to-speech and speech synthesis products were introduced (Klatt 1987). In 1983 the first commercial versions of DECTalk and Infovox synthesizers were introduced (Klatt 1987).

Some milestones of the speech synthesis history are shown in Figure 3.

Present research and commercial speech synthesis technologies involve data driven methods and algorithms from mathematics, statistics and linguistics that can easily be implemented in computers. These are rule systems (finite state rules, context free rules, rewrite rules), hidden Markov models (HMM), Neural Networks, Classification and regression trees (CART), etc.

HMMs have been applied to speech recognition from the late 1970s. For speech synthesis systems it has been used for about two decades. Neural networks have been applied in speech synthesis for about ten years. Like hidden Markov models, neural networks are also used successfully with speech recognition (Schroeder 1993).

The most promising and popular trend of present research is corpus based unit selection synthesis and data driven prosody modelling. The first unit



Gunnar Fant (KTH) with his OVE speech synthesizer

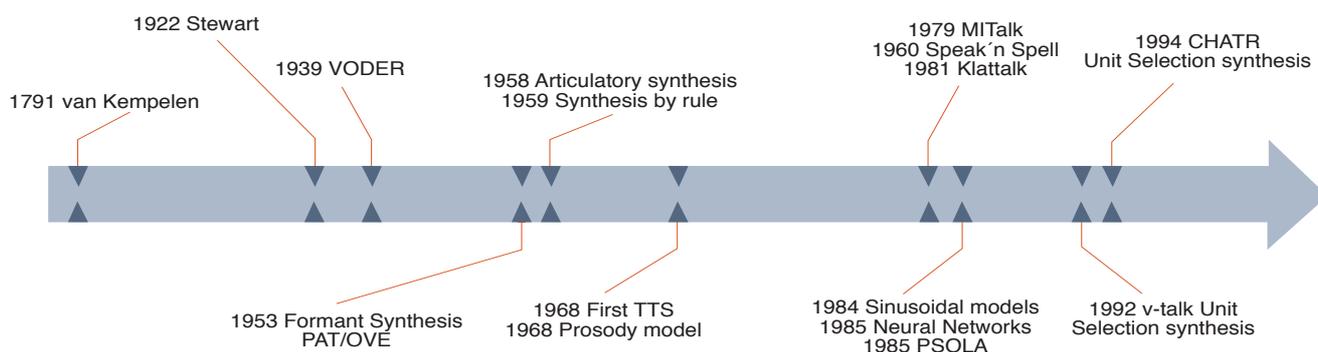
selection synthesis system was n-Talk (Sagisaka et al. 1992) and shortly after came CHATR (Black and Taylor 1994). More about unit selection synthesis and recent research in this area can be found in the *Concatenative synthesis* chapter.

Components of a Typical TTS System

It is not possible to decompose a TTS system into generic modules or components present in all TTS systems. But the boxes in Figure 4 represent high-level modules present in some way in most TTS systems. The figure can also give the impression that the different modules process data in a serial way, which is often true. But it is also common to organize the process in other ways where modules can interact both ways and in a different sequence.

There are several frameworks for encapsulating the whole TTS process to make it more easily available for users and application developers. These frameworks are briefly described in the chapter *Standard Application Frameworks – APIs and Markup Languages*. Here we will focus on the core modules of TTS where we look at generic text as input and speech in some format as output. The presentation in this chapter is inspired by the book *An introduction to text-to-speech synthesis* by Thierry Dutoit (1997). A more in-depth introduction to this field can be found there.

Figure 3 Some milestones in the history of speech synthesis



Text Pre-Processing

The first step in TTS is to convert all non-alphabetical characters, numbers, abbreviations, and acronyms into word like units. A number like “234” is converted to “two hundred and thirty four”. The beginning and end of sentences must also be detected. Text pre-processing can be done with simple lookup tables and regular grammar rules, but in some cases additional higher-level information is needed.

The same character used as sentence period can also occur as decimal point in numbers (“1.23”), in dates (“9.11.01”), and as part of abbreviations (“etc.”) and acronyms (“N.T.T.”). To solve this ambiguity you need to look at the context of the period character. But even with a context analysis you cannot solve all ambiguities. A common problem is misinterpretation of abbreviations as end of a sentence:

“Mary lives at St. Johns Street”.

To find out there is no sentence end after “St.” some higher-level analysis is needed.

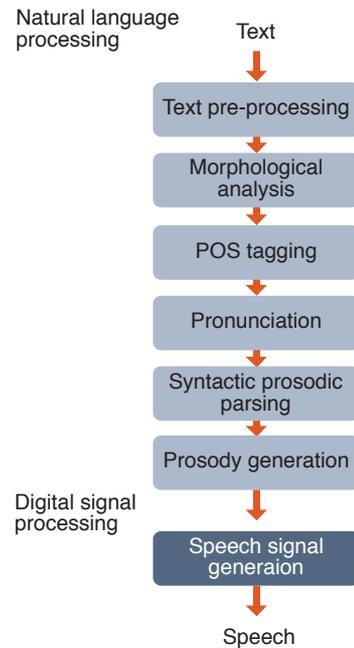
Short abbreviations can often be ambiguous. “St.” could mean street, saint or station. Number expressions such as 3/9 could be either a date or a fraction. A simple method of handling such ambiguities is to treat them in a neutral way without making a decision of what they really means. You just read 3/9 as “three slash 9” and “St” as “s t” and so on. But it is not possible to handle sentence end ambiguities in this way. A more ambitious solution to this problem is to do higher-level analysis in this or a later phase of the text analysis. Liberman and Church have proposed a probabilistic solution to the problem (Liberman & Church 1992).

Morphological Analysis

The morphological analysis module generates in principle all possible syntactic categories like noun, verb, adjective/adverb, preposition etc. and subcategories of these, for each individual word. These categories are often called Part Of Speech (POS).

The basis of the analysis is the spelling of the word, a lexicon and a morphological rule system. The lexicon contains among other things the possible POS-tags of words.

If the word is not found in the lexicon, a morphological analysis is performed trying to find the possible POS tags from decomposing the word into a stem found in the lexicon and an affix indicating that this is a particular inflectional or derivational form. A compound analysis is also often performed to find out if the word could be a compound of words found in the lexicon.



If the morphological rules for a particular language are simple, it is common to use ad-hoc rule systems, but for more complex tasks, several different morphological rule formalisms have been applied. Some of them are Definite Clause Grammars (Traber 1993), Finite State Automata (Pachunke et al. 1994) and constraint rules (Karlson 1990).

Because of the low cost of computer memory, it is now common to use large word lexica (> 100,000 words) in TTS. Typically are all the functional words (determiners, pronouns, prepositions and conjunctions) found in the lexicon because they are limited in number.

All common content words (nouns, verbs, adjectives and names) are also often included in the lexicon. But it is impossible to include all words, because new words are constructed every day, and in many languages you can have very complex derivational and inflectional word forms. The analysis must also handle misspelled words.

Context Analysis (POS Tagging)

The morphological analysis will result in one or more POS-tag for each word. A word with more than one POS-tag is ambiguous, and we call the alternatives homographs (e.g. “lives” can be both noun and verb).

You have to examine their syntactic context to select the right POS-tag. An automated POS-tagging module usually does this.

Homographs can have the same pronunciation or they can have different pronunciation. In TTS the homographs with different pronunciation are most important, because if you select the wrong

POS-tag the pronunciation will be wrong (e.g. “address”).

Several methods have been successfully applied in automated POS tagging. These are briefly categorized as deterministic taggers and probabilistic taggers. Both of them rely on the assumption that a local context analysis is sufficient in most cases for determining the right POS. This assumption makes the analysis much less complex and gives good results.

Deterministic taggers are rule-based taggers. The rules can be a collection of yes/no rules that are exploited in sequence to find the best combination of POS tags. The most well-known and efficient algorithms in this category are *decision lists* (Yarowsky 1994) and *transformation-based error-driven learning* (Brill 1994).

Another deterministic approach is based on local syntactic analysis where human experts generate syntactic rule sets for disambiguation. It takes a lot of time to construct this kind of rules and the rules are language dependent, but such POS tagging methods have also been demonstrated to do a good job (Larreur et al. 1989, Heggveit 1996). Karlson (1990) has formalized a variant of this method called Constraint Grammar. The Constraint Grammar method also has some advantages because of its ability to handle non-local dependencies.

Probabilistic taggers proposed in the literature use N-grams, Hidden Markov Models or Neural Networks.

N-gram taggers calculate the probability of a given sequence of tags. The best tag for a given word is determined by the probability that it occurs with the n previous tags. In practice $n = 3$ has shown to be sufficient for producing good results (Church 1988).

POS taggers based on Hidden Markov Models use tag sequence probabilities and word frequency measurements. The assumptions underlying this model are that each hidden tag state produces a word in the sentence, and each word is uncorrelated with all the other words and their tags and the word is depending on the n previous tags only (Kupiec 1992).

When training Neural Networks for POS tagging word sequences are presented at the input of a multi-layer perceptron and the correct POS of the word in question is given as the output. A back-propagation algorithm is used for optimising the weights (Benello et al. 1989).

It is difficult to compare the different approaches because they are seldom tested on the same test

set. But, rule based approaches, trigrams and HMM models have all shown to produce good results (95–97 % correctly tagged), but they all have their different advantages and disadvantages.

Pronunciation

In early TTS systems it was common to build rule systems (grapheme-to-phoneme rules) that covered the pronunciation of most common words, and have an exceptions dictionary for the words that did not get the right pronunciation from the rules. This approach was motivated from the fact that a large pronunciation lexicon was expensive in terms of memory requirements.

Most modern TTS systems include a large word lexicon including pronunciation and possibly different style and dialect variations. But even with a large word lexicon you need rules for the pronunciation of words not found in the lexicon and rules for the pronunciation of words analysed as compounds or derivational variants of words found in the lexicon.

Most of the proposed systems are expert rule-based systems, but there are also automatically trained rule-based systems like decision tree models (Lucassen and Mercer 1984), Hidden Markov Models (Van Coile 1991), pronunciation by analogy (Yvon 1996), stochastic phonographic transduction (Luk and Damper 1996), and Neural Network based rule systems like NETSPEAK (McCulloch et al. 1987). The best of these automatically trained rule-based systems can compare in performance to carefully designed manual rules.

The pronunciation of proper names is more complex than ordinary words. There are far more proper names than ordinary words, and their origin can be from very different languages. Pronunciation rules made for one language is not very useful for pronunciation of foreign proper names. Some kind of etymological classification procedure is needed as the first step in rules for finding the pronunciation of foreign names (Church 1986). In the European project ONOMASTICA proper name pronunciation lexica for twelve European languages were developed (Schmidt et al. 1993).

Prosody Generation

Prosody is properties of the speech signal related to intonation (pitch), timing (segment duration) and intensity. Pitch and duration are considered to be the most important factors.

Intonation Modelling

The pitch pattern or fundamental frequency (f_0) over a sentence in natural speech is a combination of many factors. The pitch contour depends on the meaning of the utterance. For example,

in normal speech the pitch slightly decreases toward the end of the utterance and when the utterance is a yes/no question, the pitch pattern will rise at the end of the utterance. In the end of an utterance there may also be a continuation rise, which indicates that there is more speech to come. A raise or fall in fundamental frequency can make a syllable stand out within an utterance and thereby make a word or phrase more prominent in an utterance context. The most prominent phrase in an utterance is called the focus of the utterance.

Prosodic and intonational features are also involved in grouping of syllables and words into larger chunks that are not necessarily identical to syntactic structures.

The pitch contour is also affected by gender, physical and emotional state, and attitude of the speaker.

It is common to study and model at least two different levels of intonation in TTS synthesis.

- A linguistic model represents the intonation of an utterance as a sequence of abstract units that have some function in speech communication.
- An acoustic model represents the acoustic fundamental frequency contour

Acoustic models aim to present a fundamental frequency contour in a compact way.

Fujisaki's model (Fujisaki 1992) represents a fundamental frequency contour as a series of phrase and accent commands realized as pulses and step functions filtered through second order linear filters. The summed output yields the f_0 contour. Good approximations of f_0 contours have been obtained for several languages.

F_0 contours can also be modelled as a sequence of target points typically highs and lows, with some interpolation function filling the contour gap between the target points. The RFC model (Taylor 1994) and the *quadratic splines method* (Hirst et al. 1991) are examples of this model strategy.

Although the parameters of acoustic intonation models can be automatically predicted from text and syntactic prosodic information, it is commonly accepted, at least among linguists, that a separate linguistic or phonological intonation model is a useful representation level prior to the acoustic level of predicting intonation.

Linguistic models of intonation define a number of phrase- or clause-like units and an intonation

grammar to account for the structure of utterances in terms of such intonation units.

Selkirk (1984) presents a phonological hierarchy containing syllable, foot, prosodic word, phonological phrase and intonational phrase. Variations of this hierarchy are applied in several models and for several languages, but no universal model of prosodic units and syntax has been established.

A model of American English intonation originally developed by Pierrehumbert (1980), and recently known as the ToBI model (Silverman et al. 1992) represents the intonation as sequences of H (high) and L (low) tones and combinations of these associated with the units in a phonological hierarchy similar to the one presented by Selkirk.

ToBI-like models have been successfully applied to many other languages, for example Japanese (Pierrehumbert and Beckman 1988) and Swedish (Bruce and Granström 1993). *The Trondheim model of intonation* (Nilsen 1992) is the preferred model for phonological analysis of Norwegian in the Norwegian prosody research community.

Syntactic-Prosodic Parsing

In prediction of prosody we need to predict some prosodic units. The units can be those from a specific linguistic model, or simpler ad-hoc melodic phrases.

It is well known that syntax plays an important role in predicting prosody but the meaning of an utterance (semantics) and other factors are also involved. Prosodic phrases often coincide with syntactic phrases. Although a perfect prosody model must account for all the possible input factors that humans use in selecting the right prosody, we can still get a fairly good neutral prosody model without taking complex syntax, semantics and context knowledge into account. Such structuring of text for prosody generation purposes is often referred to as syntactic-prosodic parsing.

Several very different syntactic-prosodic parsing methods are reported.

The *chinks'n chunks* algorithm proposed by Liberman and Church (1992), is a simple but effective heuristic rule-based method. A minor prosodic phrase is defined as a sequence of chinks followed by a sequence of chunks, where chinks are function words and tensed verb forms, and chunks are content words and object pronouns. Simple rules for finding the major phrases of a sentence can be defined from punctuation and specific words indicating clause boundaries.

Another and more ambitious method is to use traditional syntax parsing techniques to find more complete syntactical structures in a sentence, and map such structures to relevant prosodic structures. Definite Clause Grammars (DCG) and Unification Grammars (UG) are popular context-free grammar frameworks for general text syntax parsing. One of the most ambitious DCG parser implemented in a TTS system is the one reported by Traber (1993). A large grammar of German language is implemented and augmented with penalties for each grammar rule. The parse tree with the lowest total accumulated penalty is chosen as the resulting parse. The mapping to prosodic phrase structure is done directly by mapping strong syntactic boundaries to major prosodic phrase boundaries and lower level syntactic boundaries to minor phrases. The number of prosodic phrase boundaries is reduced by not allowing too close phrase boundaries.

This kind of direct mapping from syntax to prosody will often fail (Bachenko and Fitzpatrick 1990) and the resulting prosodic phrasing can in some cases be worse than no phrasing.

Corpus based methods are also applied in syntactic-prosodic parsing. Large text corpora labelled with prosodic phrase boundaries and other model specific boundaries can be used as input for training models to predict the same boundaries. Classification and regression tree (CART) (Breiman et al. 1984) is a popular method for this purpose, and CART techniques have been applied by among others Wang and Hirschberg (1991).

Fackrell et al. (1999) use CART and neural nets for modelling prosody in six different European languages. This work and others represent a new trend in prosody research towards corpus based and language independent prosody models. This is inspired by the success of corpus based methods in other speech technology areas and the commercial need for building TTS systems that handle many different languages effectively in one single framework.

Duration Modelling

The duration or timing characteristics can be analysed at several levels from phoneme (segmental) durations to sentence level timing, speaking rate, and rhythm.

Although theoretical duration and speech rhythm studies are often based on syllable or stress units, most TTS systems focus on the duration of phone segments. The influence of higher-level units is incorporated as various input factors to a phonetic duration model. In general, the phoneme duration depends on the phoneme

itself, neighbouring phonemes, syllable position, syllable stress, word and phrase boundaries, speech rate and rhythm.

A number of different models have been proposed, and most of them use linguistic knowledge or exploratory statistics to limit the model complexity and number of input parameters.

Additive and multiplicative rule models are based on an intrinsic duration modified by additive or multiplicative factors (Bartkova and Sorin 1987).

van Santen's duration model is based on a combination of multiplicative and additive rule sets for different classes of phonemes, where parameters are estimated from statistical analysis of large speech corpora (van Santen 1994).

Purely statistical approaches are based on Classification And Regression Trees (CART) (Riley 1992) or neural networks (Campbell 1992).

Signal Generation

Synthesized speech can be produced by several different methods. The methods can be classified into three groups:

- Articulatory synthesis, which attempts to model the human speech production system;
- Formant synthesis, which models the transfer function of the vocal tract based on a source-filter-model;
- Concatenative synthesis, which uses different length pre-recorded samples derived from natural speech.

The first successful commercial TTS systems were based on formant synthesis, and this method was dominant during the 70s and 80s, and still there are several commercial systems applying formant synthesis. During the 90s, concatenative synthesis became the method of choice in commercial speech synthesis. Articulatory synthesis still gets considerable attention in the research community.

Articulatory Synthesis

Articulatory synthesis tries to model the human vocal organs. It is an intuitively attractive method and it can potentially produce high-quality synthetic speech. Another great advantage is that it requires much fewer parameters to describe a speech sound.

On the other hand, it is also one of the most complex methods to implement and the computational load is also considerably higher than with other methods. It has not yet achieved the



same level of success as the other methods, but as computational power gets cheaper, articulatory synthesizers become more attractive (Boersma 1998).

Articulatory synthesis involves models of the human articulators and vocal cords. The articulators are usually modelled with a set of area functions between glottis and mouth. The first articulatory model was based on a table of vocal tract area functions from larynx to lips for each phonetic segment (Klatt 1987).

When speaking, the vocal tract muscles cause articulators to move and to change shape of the vocal tract, which causes different sounds. The data for an articulatory model is derived from X-ray, MRI and CT (Computer Tomography) analysis of natural speech (Engwall 1999).

Formant Synthesis

The most widely applied synthesis method during the last decades has been formant synthesis, which is based on a source-filter-model of speech where the formants are the resonance frequencies of the filter. In general there are two basic structures, parallel and cascade. For better performance some kind of combination of these is usually used. Formant synthesis is a flexible synthesis method because it can model very different voices just by manipulating the input parameters. The biggest problem is to control all the parameters detailed enough to produce natural sounding speech. A formant synthesizer requires rules

where each sound or sound in a context is associated with a sequence of input parameters.

At least three formants are generally required to produce intelligible speech and up to five formants to produce high quality speech. Each formant is usually modelled with a two-pole resonator, which enables both the formant frequency (pole-pair frequency) and its bandwidth to be specified (Donovan 1996).

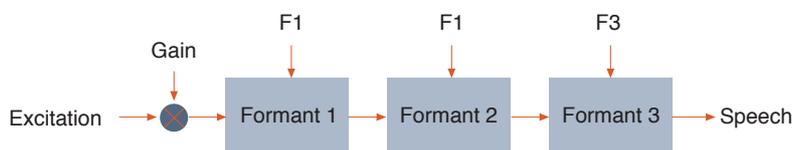
A cascade formant synthesizer consists of band-pass resonators connected in series and the output of each formant resonator is applied to the input of the following one. The cascade structure needs only formant frequencies as control information. The main advantage of the cascade structure is that the relative formant amplitudes for vowels do not need individual controls (Allen et al. 1987).

The cascade structure has been found better for non-nasal voiced sounds and because it needs less control information than parallel structure, it is then simpler to implement. However, with cascade model the generation of fricatives and plosive bursts is a problem.

A parallel formant synthesizer consists of resonators connected in parallel. Sometimes extra resonators for nasals are used. The excitation signal is applied to all formants simultaneously and their outputs are summed. The parallel structure enables controlling of bandwidth and gain for each formant individually and thus needs also more control information.

The parallel structure has been found to be better for nasals, fricatives, and stop-consonants, but some vowels cannot be modelled with parallel formant synthesizer as well as with the cascade one. In 1980 Dennis Klatt (Klatt 1980) proposed a more complex formant synthesizer, which incorporated both the cascade and parallel synthesizers with additional resonances and anti-resonances for nasalized sounds, a sixth formant for high frequency noise and a bypass path to give a flat transfer function. The system used a complex excitation model which was controlled by 39 parameters updated every 5 ms. The Klatt Formant Synthesizer was incorporated into several TTS systems, such as MITalk and DECtalk. The latest improvement of this formant synthesizer with a reduced set of input parameters is called HLSyn (Hanson et al. 1997).

Figure 5 Basic structure of cascade formant synthesizer (Fn: Formant frequencies)



Concatenative Synthesis

Concatenation of pre-recorded natural speech phrases is not a new synthesis method. Simple variants of this are well known from telephone services like “talking clock” etc. It is probably the easiest way to produce intelligible and natu-

ral sounding synthetic speech. However, concatenative synthesizers are less flexible than other methods. It is usually limited to one speaker and one voice and requires more memory capacity than other methods.

We can classify concatenative synthesis in two main categories, fixed unit synthesis and unit selection synthesis.

Fixed Unit Synthesis

Fixed unit synthesis was the first concatenative synthesis method that became popular in TTS systems in the late 80s. One of the most important aspects in fixed unit concatenative synthesis is to find the best unit length. The selection is usually a trade-off between longer and shorter units. With longer units high naturalness, less concatenation points and good control of coarticulation is achieved, but the amount of units and memory required is increased. With shorter units, less memory is needed, but the sample collecting and labelling procedures become more complex.

Most fixed unit TTS systems are based on using diphones, demisyllables or combinations of these. Demisyllables represent “half-syllables”, the initial part of a syllable or the final part of a syllable. One advantage of demisyllables is that only about 1,000 of them are needed to construct the 10,000 syllables of English (Donovan 1996). Using demisyllables, instead of diphones, requires considerably less concatenation points. Demisyllables also contain most of the natural voice transitions and coarticulation effects from the recorded speech. However, the memory requirement was considered to be high.

Diphones are defined as units that begin in the middle of a steady state part of a phone to the middle of the following one, so they contain the transitions between adjacent phones. That means that the concatenation point will be in the most steady state region of the signal, which reduces the distortion from concatenation points. In principle, the number of diphones is the square of the number of phonemes (plus allophones), but not all combinations of phonemes are needed. The number of units is usually from 1500 to 2000. Because of this relatively small number of units together with other advantages, diphones are considered to be a very suitable unit for TTS synthesis.

A speech segment database for fixed unit synthesis is constructed by first defining a complete list of units needed for synthesizing all common sound combinations in a language. Then a corresponding list of carrier words containing the units at least once is completed. The units should be placed in a neutral context in each carrier

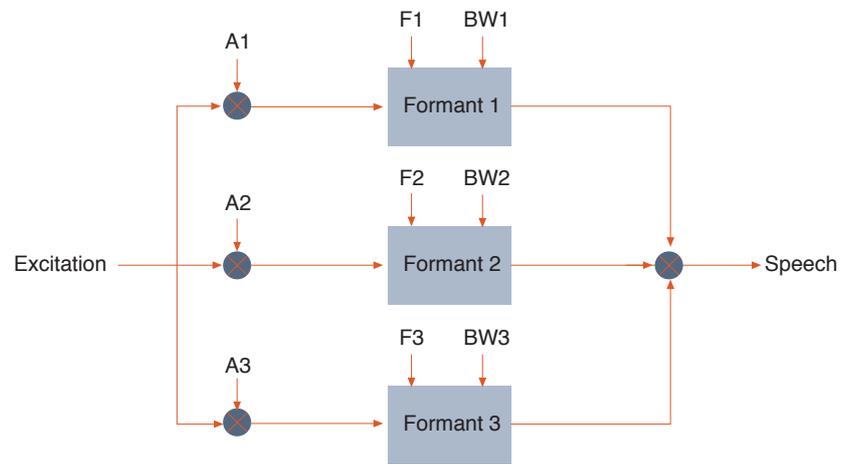


Figure 6 Basic structure of a parallel formant synthesizer. (An: amplitude, Fn: formant frequency, BWn: formant bandwidth)

word to avoid coarticulation and other word specific effects. When recording the database there must also be some control with pitch level, reading speed and amplitude to ensure that the units are read approximately at the same level.

When recorded, the speech database is automatically or semi-automatically segmented, and the synthesis units are extracted, stored in a database and parameterized for easy retrieval at synthesis run-time.

Unit Selection Synthesis

In unit selection synthesis, the concatenated units are not selected until synthesis run-time. A large speech corpus is used as the acoustic unit inventory, and the sequence of segments from the corpus that best match the target sequence is selected. The first laboratory TTS system based on unit selection was n-Talk (Sagisaka et al. 1992), and later ATR’s CHATR system (Black and Taylor 1994). By the end of the 90s unit selection became a main topic for speech synthesis research, and several commercial systems are introduced (IBM Concatenative Text-to-Speech, AT&T NextGen, Rhetorical rVoice). Designing a high quality unit selection synthesis system involves many steps including the selection of concatenation unit(s), selection of distance measure and weight training method, corpus design and database preparation.

Unit Length

A main advantage in unit selection synthesis is the reduction in number of concatenation points and reduced concatenation distortion, compared to fixed unit synthesis. A concatenation is always a potential source of speech signal distortion originating from the concatenation itself and the signal processing involved. The basic units can still be small and uniform, like diphones, phones and half-phones. Because of the large number of units to choose from in the synthesis corpus, the concatenation distortion between units can be small, or even zero if the selected

units are found in sequence in the synthesis corpus. The actual length of concatenated units can in this way be longer than the basic unit length. The main advantage of using small uniform length units is the flexibility to construct longer units from shorter basic units, when not finding a perfect match for the longer units. The selection of basic unit length is a trade-off between this flexibility and the corresponding complexity of selecting the right unit.

In a pure unit selection system the units are simply concatenated and played back. If the whole utterance is found in the corpus, the whole synthesis process reduces to a simple playback of this utterance. On the other hand, if the target utterance is constructed from a sequence of small units from synthesis corpus, the same problems can occur as in fixed unit synthesis.

Distance Measure

A unit selection algorithm attempts to minimize two types of cost or distant measure, one for unit distortion and one for continuity distortion (Möbius 2000). Unit distortion is a distance measure between the target unit and the candidate unit, and continuity distortion is a distance measure between two units at the concatenation point.

An algorithm proposed by Hunt and Black (1996) selects an optimal sequence of units by finding the path through a state transition network that minimizes the combined target and concatenation costs, where each unit is a state in the network. To train the feature weights of the model, they propose independent training of the two cost functions using multiple linear regression. Recent improvements of this regression training method (Meron and Hirose 1999) makes it possible to train the two costs simultaneously and taking the cost of prosodic modification into account.

The target cost is an estimated distance between vectors of segmental and prosodic target values like phonemic context, syllable context, accent and phrase position and other model specific data generated by the linguistic processing module of a TTS system.

The concatenation cost is an acoustic distance between two units at the concatenation point. The acoustic distance measure can be mel-based cepstral distance, but it is still a subject for further research to find measures that match the perceptual distance better. Concatenation cost can be estimated for all possible unit concatenations offline and stored in a database because all the input data needed are already present in the corpus. This will reduce the computational needs at run-time, but the storage cost can be significant.

A different unit selection alternative is to use phonologically based unit selection, avoiding detailed acoustic constraints. The idea is that the acoustics will be appropriate if the units are selected from a phonologically matching context. Variations of this method have been proposed in the literature (Breen and Jackson 1998, Taylor and Black 1999).

Corpus Design and Database Preparation

In unit selection synthesis, the unit inventory corpus is very important. A well-designed corpus can be the difference between a really good sounding synthesis and a bad one. The size of the corpus is dependent on its design and on the type of units involved in the synthesis.

If the utterances are randomly selected, the corpus must be bigger than if the utterances are selected according to some coverage criteria. A common method is to use a greedy algorithm to refine the corpus by iteratively selecting utterances that give the best total phonetic and prosodic coverage.

A corpus for small size unit selection can also potentially be smaller than a corpus for bigger size unit selection because of the flexibility to construct bigger units from smaller units when the target and concatenation cost is low.

A unit inventory corpus can never cover all units in all contexts. There will always be candidate units and concatenations that match poorly. To gently modify unit prosody and to smooth non-matching concatenations, a signal modification technique is useful.

The sound of unit concatenation synthesis will be dependent on the speaking style of the person reading the corpus and on the corpus itself. A news reporter reading a news corpus will make the synthesis sound like news reading, and a poet reading a poetry corpus will make the synthesis sound like poetry. The application of the synthesis must be taken into account when selecting text content and speaking style.

The corpus must be segmented into units relevant to the units selected, and the corpus must be labelled with the relevant acoustic, phonetic and prosodic features applied in estimating target and concatenation costs. Automatic segmentation and labelling can be done applying Hidden Markov Models iteratively to improve the segmentation. When segmented, each unit can be labelled with the phoneme identity, duration, pitch contour, energy, etc.

Prosodic labelling is needed if prosodic features are taken into account when estimating the target cost. A particular prosodic framework must be

chosen and the labelling must be done according to this. Automatic prosodic labelling is reported to have potential for being as good as, or even better than manual labelling (Syrdal et al. 2000).

For a more in-depth overview of unit selection synthesis, see Möbius (2000).

Signal Modification Techniques

PSOLA

The PSOLA (Pitch Synchronous Overlap Add) method was originally developed at France Telecom (CNET). It is not a synthesis method itself but allows pre-recorded speech samples to be smoothly concatenated and provides a method for modifying pitch and duration. This method is used in several commercial synthesis systems, such as Elan (www.elan.com).

There are several versions of the PSOLA algorithm. The time-domain version, TD-PSOLA, is the most commonly used due to its simplicity and computational efficiency. The basic algorithm consists of three steps (Moulines and Charpentier 1990):

- The analysis step where the original speech signal is divided into separate overlapping short-term analysis signals (ST);
- The modification of each short-term analysis signal;
- The synthesis step where these segments are recombined by means of overlap and add.

Short term signals $x_m(n)$ are obtained from digital speech waveform $x(n)$ by multiplying the signal by a sequence of pitch-synchronous analysis window $h_m(n)$:

$$x_m(n) = h_m(t_m - n)x(n)$$

where m is an index for the short-time signal.

The windows are centred on the successive instants t_m , called pitch-marks. These marks are set at a pitch-synchronous rate in the voiced parts of the signal and often at a constant rate in the unvoiced parts. But to avoid synthetic voicing of unvoiced sounds it is necessary to modify the constant rate by a random component. The used window length is proportional to local pitch period and the window factor is usually from 2 to 4 (Charpentier 1989). The pitch markers are determined automatically or semi-automatically by some pitch estimation method. The segment recombination in synthesis step is performed after defining a new pitch-mark sequence.

Manipulation of fundamental frequency is achieved by changing the time intervals between pitch markers (see Figure 7). The modification of duration is achieved by either repeating or omitting speech segments.

Other variations of PSOLA, Frequency Domain PSOLA (FD-PSOLA) and the Linear-Predictive PSOLA (LP-PSOLA), are theoretically more appropriate approaches for pitch-scale modifications because they provide independent control over the spectral envelope of the synthesis signal (Moulines and Charpentier 1990). MBROLA (Dutoit 1997) makes use of a diphone database obtained after a Harmonic/Stochastic analysis-synthesis of an original diphone database. The resulting synthesis technique takes advantage of the flexibility of parametric speech models while keeping the computational simplicity of time-domain synthesizers.

Linear Prediction Based Methods

Linear predictive (LP) methods are originally designed for speech coding systems, but may also be used in speech synthesis. Like formant synthesis, the basic LP is based on the source-filter-model of speech described earlier. The digital filter coefficients are estimated automatically from a frame of natural speech.

The basis of linear prediction is that the current speech sample $y(n)$ can be approximated or predicted from a finite number of previous p samples $y(n-1)$ to $y(n-k)$ by a linear combination with small error term $e(n)$ called residual signal.

The main deficiency of the ordinary LP method is that it represents an all-pole model, which means that nasals and nasalized vowel phonemes are poorly modeled. The quality is also poor with short

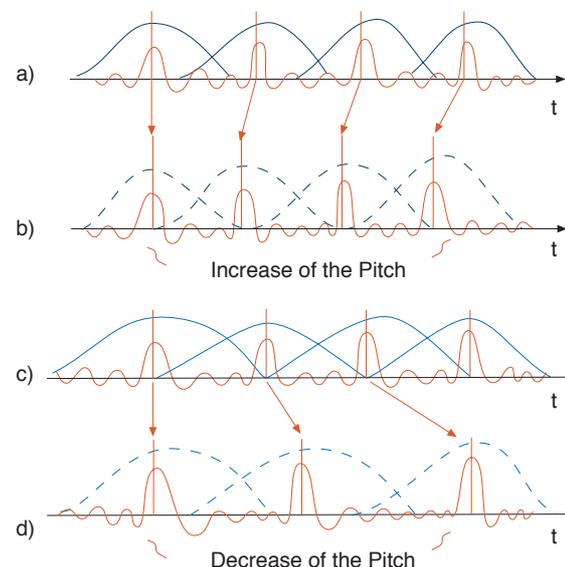


Figure 7 Pitch modification of a voiced speech segment

plosives because the time-scale events may be shorter than the frame size used for analysis. With these deficiencies the speech synthesis quality with standard LP method is generally considered poor.

Variations of linear prediction have been developed to increase the quality of the basic method (Donovan 1996). With these methods the used excitation signal is different from ordinary LP methods and the source and filter are no longer separated. These kinds of variations are for example multipulse linear prediction (MLPC) where the excitation is constructed from a set of several pulses, residual excited linear prediction (RELP) where the error signal or residual is used as an excitation signal and the speech signal can be reconstructed exactly, and code excited linear prediction (CELP) where a finite number of excitations used are stored in a finite codebook.

Sinusoidal Models

Sinusoidal models are based on the assumption that the speech signal can be represented as a sum of sine waves with time-varying amplitudes and frequencies (Macon 1996, Kleijn et al. 1998). In the basic model, the speech signal $s(n)$ is modelled as the sum of a small number L of sinusoids

$$s(n) = \sum_{l=1}^L A_l \cos(\omega_l n + \phi_l)$$

where $A_l(n)$ and $\phi_l(n)$ represent the amplitude and phase of each sinusoidal component associated with the frequency track ω_l . To find these parameters $A_l(n)$ and $\phi_l(n)$, the DFT of windowed signal frames is calculated, and the peaks of the spectral magnitude are selected from each frame. The basic model has also some modifications such as ABS/OLA (Analysis by Synthesis / Overlap Add) and Hybrid / Sinusoidal Noise models (Macon 1996).

While the sinusoidal models are suitable for representing periodic signals, such as vowels and voiced consonants, the representation of unvoiced speech is problematic (Macon 1996).

Standard Application Frameworks – APIs and Markup Languages

A Speech Application Programming Interface (SAPI) is an interface between applications and speech technology engines. The interface allows multiple applications to share the available speech resources on a computer without having to program different proprietary speech engine APIs. The user of an application can also choose from a list of synthesizers as long as it supports a SAPI. Currently SAPIs are available for several environments, such as MS-SAPI for Microsoft Windows (www.microsoft.com/speech) and Sun Microsystems Java SAPI (JSAPI) for JAVA

based applications (<http://java.sun.com/products/java-media/speech/>).

The Speech Synthesis Markup Language (SSML) Specification is part of the W3C Voice Browser Working Group set of new markup specifications for voice browsers (<http://www.w3.org/Voice/>), and is designed to provide an XML-based markup language for assisting the generation of synthetic speech in web applications and other applications. The role of the markup language is to provide authors of synthesisable content a standard way to control aspects of speech such as pronunciation, volume, pitch, rate, etc. across different synthesis-capable platforms.

The SSML standard is also part of the W3C VoiceXML 2.0 dialogue markup specification, and the recent dialogue specification proposed by Microsoft and others (www.saltforum.org).

References

- Bachenko, J, Fitzpatrick, E. A Computational Grammar of Discourse-Neutral Prosodic Phrasing in English. *Computational Linguistics*, 16, 155–167, 1990.
- Bartkova, K, Sorin, C. A Model of Segmental Duration for Speech in French. *Speech Communication*, 15 (1-2), 127–139, 1987.
- Black, A, Taylor, P. CHATR : A Generic Speech Synthesis System. *COLING94*, Japan, 1994.
- Boersma, P. *Functional Phonology [LOT International Series 11]*. The Hague, Holland Academic Graphics, 1998. Doctoral thesis, University of Amsterdam.
- Breen, A P, Jackson, P. Non-uniform unit selection and the similarity metric within BT's Laureate TTS system. In: *Proceedings of the Third ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia, 373–376, 1998.
- Breiman, L et al. *Classification and regression trees*. Monterey, CA, Wadsworth & Brooks, 1984.
- Benello, J et al. Syntactic Category Disambiguation with Neural Networks. *Computer Speech and Language*, 3, 203–217, 1989.
- Brill, E. Some Advances in Transformation-based Part of Speech Tagging. *Proceedings of the AAAI'94*, 722–727, 1994.
- Bruce, G, Granström, B. Prosodic Modelling in Swedish Speech Synthesis. *Speech Communication*, 13, 63–73, 1993.

- Campbell, W N. Syllable-based segmental duration. In: G Bailly, C Benoit, T Sawallis (eds.). *Talking machines : Theories, models, and designs*, 211–224. Elsevier, 1992.
- Carlson, R, Granström, B, Nord, L. Evaluation and Development of the KTH Text-to-Speech System on the Segmental Level. *Proceedings of ICASSP 90*, (1), 317–320, 1990.
- Church, K W. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. *Proceedings of the Second Conference on Applied Natural Language Processing*, Austin, Texas, 1988.
- Donovan, R. 1996. *Trainable Speech Synthesis*. Cambridge University Engineering Department, England. (PhD thesis) <ftp://svr-ftp.eng.cam.ac.uk/pub/reports/donovan_thesis.ps.Z>
- Dutoit, T. *An introduction to text-to-speech synthesis*. Dordrecht, The Netherlands, Kluwer Academic Publishers, 1997. (ISBN 0-7923-4498-7)
- Engwall, O. Modelling of the vocal tract in three dimensions. *Proc of Eurospeech 1999*, 113–116, 1999.
- Fackrell, J W A. Multilingual prosody modelling using cascades of regression trees and neural networks. *Proceedings of Eurospeech 99*, 1999.
- Fant, G, Martony, J. *Speech Synthesis*. Stockholm, Sweden, Speech Transmission Laboratory, Royal Institute of Technology, QPSR 2, 18–24, 1962.
- Flanagan, J. *Speech Analysis, Synthesis, and Perception*. Berlin-Heidelberg-New York, Springer-Verlag, 1972.
- Fujisaki, H. The role of Quantitative Modeling in the Study of intonation. *Proceedings of the International Symposium on Japanese Prosody*, 163–174, 1992.
- Hanson, H M et al. New parameters and mapping relations for the Hlsyn speech synthesizer. *J. Acoustical Soc. America*, 102, 3163, 1997.
- Heggtveit, P O. A Generalized LR Parser for Text-to-speech Synthesis. In: *Proceedings of ICSLP '96*, 3, 1429–1432, 1996.
- Hirst, D J et al. Coding the F0 of a continuous text in French : An experimental Approach. *Proceedings of the International Congress of Phonetic Sciences*, Aix en Provence, 234–237, 1991.
- Hunt, A J, Black, A W. Unit selection in a concatenative speech synthesis system using a large speech database. In: *Proceedings of the ICASSP 96*, Munich, 1, 373–376, 1996.
- Karlson, F. Constraint Grammars as a Framework for Parsing Running Text. *Proceedings of the Conference on Computational Linguistics*, Helsinki, 3, 168–173, 1990.
- Klatt, D. Software for a Cascade/Parallel Formant Synthesizer. *Journal of the Acoustical Society of America, JASA*, 67, 971–995, 1980.
- Klatt, D H. Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, 82, 737–793, 1987.
- Kleijn, K, Paliwal, K (eds.). *Speech Coding and Synthesis*. The Netherlands, Elsevier, 1998.
- Kupiec, J. Robust Part-of-Speech Tagging Using a Hidden Markov Model. *Computer Speech and Language*, 6, 225–242, 1992.
- Kurzweil, R. *The Age of Intelligent Machines*. Cambridge, MA, MIT Press, 272–281, 1990.
- Larreur, D et al. Linguistic and Prosodic Processing for a Text-to-Speech Synthesis Systems. *Proceedings of Eurospeech 89*, Paris, 510–513, 1989.
- Lieberman, M Y, Church, K W. Text Analysis and Word Pronunciation in Text-to-Speech Systems. In: *Advances in Speech Signal Processing*. S Furui, M Sondhi (eds.). New York, Dekker, 791–831, 1992.
- Lucassen, J M, Mercer, R L. An Information Theoretic Approach to the Automatic Determination of Phoneme Base Forms. *Proceedings of ICASSP 84*, 52.5, 42–45, 1984.
- Luk, R W P, Damper, R I. Stochastic Phonographic Transduction for English. *Computer, Speech and Language*, 10, 133–153, 1996.
- Macon, M. *Speech Synthesis Based on Sinusoidal Modeling*. Georgia Institute of Technology, 1996. Doctorial Thesis.
- McCulloch, N et al. NETSPEAK – a re-implementation of NETTALK. *Computer, Speech and Language*, 2, 289–301, 1987.
- Meron, Y, Hirose, K. Efficient weight training for selection based synthesis. In: *Proceedings of Eurospeech99*, Budapest, Hungary, 5, 2319–2322, 1999.
- Moulines, E, Charpentier, F. Pitch-synchronous Waveform Processing Techniques for Text-To-

- Speech Synthesis Using Diphones. *Speech Communication*, 9, 453–467, 1990.
- Möbius, B. Corpus-Based Speech Synthesis: Methods and Challenges. *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung, AIMS*, 6 (4), 87–116, 2000.
- Nebbia, L et al. A Specialized Speech Synthesis Technique for Application to Automatic Reverse Directory Service. *Proceedings of IVTTA '98*, Torino, September 1998, 223–228.
- Nilsen, R A. *Intonasjon i interaksjon : sentrale spørsmål i norsk intonologi*. Trondheim, 1992. Doctoral dissertation. (In Norwegian)
- Pachunke, T et al. The Linguistic Knowledge in a Morphological Segmentation Procedure of German. *Computer, Speech and Language*, 8, 233–245, 1994.
- Pierrehumbert, J. *The Phonology and Phonetics of English Intonation*. MIT, Indiana University Linguistics Club, 1980. PhD dissertation.
- Pierrehumbert, J, Beckman, M. *Japanese Tone Structure*. Cambridge, MA, MIT Press, 1988.
- Riley, M D. Tree-Based Modelling for Speech Synthesis. In: G Bailly, C Benoit, T Sawallis (eds.). *Talking machines : Theories, models, and designs*, 211–224. Amsterdam, Elsevier, 1992.
- Sagisaka, Y et al. ATR – n-Talk speech synthesis system. *Proceedings of ICSLP92*, Banff, Canada, 1992, 483–486.
- Schmidt, M et al. Phonetic transcription standards for European names (ONOMASTICA). *Proceedings EUROSPEECH'93*, Berlin, 1993, I, 279–282.
- Schroeder, M. A Brief History of Synthetic Speech. *Speech Communication*, 13, 231–237, 1993.
- Selkirk, E O. *Phonology and Syntax : the Relation Between Sound and Structure*. Cambridge, MA, MIT Press, 1984.
- Silverman, K et al. ToBI: a standard for labelling English prosody. *Proceedings of ICASSP 92*, Alberta, 1992, 867–870.
- Syrdal, A et al. Corpus-based techniques in the AT&T NextGen synthesis system. *Proceedings of ICSLP2000*, Beijing, 2000.
- Taylor, P. The Rise-Fall-Connection Model of Intonation. *Speech Communication*, 15, 1994.
- Taylor, P, Black, A W. Speech synthesis by phonological structure matching. *Proceedings of Eurospeech*, Budapest, Hungary, 2, 623–626, 1999.
- Traber, C. Syntactic Processing and Prosody Control in the SVOX TTS System for German. *Proceedings of Eurospeech 93*, Berlin, 3, 2099–2102, 1993.
- van Coile, B. Inductive Learning of Pronunciation Rules with the DEPES System. *Proceedings of ICASSP 91*, Toronto, 2, 745–748, 1991.
- van Santen, J P H. Assignment of segmental duration in text-to-speech synthesis. *Computer, Speech and Language*, 8, 95128, 1994.
- Wahlster, W (ed.). *Verbmobil : Foundations of Speech-to-Speech Translation*. Berlin – Heidelberg, Springer, 2000. (ISBN 3-540-67783-6)
- Wang, M Q, Hirschberg, J. Predicting Intonational Boundaries Automatically from Text : The ATIS Domain. *Proceedings of the DARPA Speech and Natural Language Workshop*, 378–383, 1991.
- Yarowsky, D. Homograph Disambiguation in Speech Synthesis. *Proceedings of the 2nd ESCA/IEEE Workshop on Speech Synthesis*, New Paltz, NY, 1994.
- Yvon, F. Grapheme-to-Phoneme Conversion of Multiple Unbounded Overlapping Chunks. *Proceedings of NemLap'96*, Ankara, 1996.

Auditory Based Methods for Robust Speech Feature Extraction

BOJANA GAJIC



Bojana Gajic (32) received her Siv.Ing. (MSc) and Dr.Ing. (PhD) degrees in telecommunications from the Norwegian University of Science and Technology (NTNU) in 1996 and 2002, respectively. She has worked as a visiting researcher in the field of automatic speech recognition at NTT Human Interface Laboratories in Tokyo (1995–96), AT&T Labs Research in New Jersey (1999–2000), and Griffith University in Brisbane (2000). She is currently a post doctoral researcher at the Department of Telecommunications at NTNU, working in the field of automatic speech recognition.

gajic@tele.ntnu.no

A major limitation for the use of automatic speech recognition in many practical applications is its lacking robustness against changing environmental noise. Automatic speech recognition is based on a sequence of speech feature vectors that contain the information relevant for discriminating between different speech sounds. One possible way to increase the robustness of speech recognition systems is to make the feature vectors less sensitive to the changes in environmental noise, while retaining their good discriminative properties.

Humans' exceptional ability to recognize speech in noise has inspired the research on robust feature extraction. Conventional feature extraction methods already incorporate some auditory-based concepts. This paper gives an overview of several alternative feature extraction methods that make use of more detailed knowledge of human speech perception. They have generally shown greater robustness in the presence of environmental noise compared to the conventional methods.

A common characteristic of the alternative methods is the use of the information about spectral peak positions, which is not sensitive to the changes in environmental noise. This is probably the major reason for their noise robustness, rather than the detailed modeling of the processes in the human auditory system.

1 Introduction

A major limitation of the state-of-the-art automatic speech recognition (ASR) systems is their lacking robustness against changes in the environmental noise. The performance of ASR systems reduces dramatically when they are used in the conditions that differ from those observed under training.

In many practical applications of ASR, the environmental noise is unknown or changing. Thus, it cannot be accounted for during system training. This is the typical situation for the use of the pocket-size mobile communication devices (e.g. mobile phones, PDAs). The use of ASR in such devices is very attractive due to their ever decreasing size.

Much research during the last decade has been concentrated towards increasing the noise robustness of ASR systems. The aim has been to reduce the mismatch between the incoming speech signal and the trained speech models. Robust feature extraction is one possible way of dealing with the problem.

The role of feature extraction in ASR is to extract from the speech signal the information relevant for discriminating between different speech sounds. Conventional speech features have good discriminative abilities on the clean speech. However, they are not invariant to changes in the environmental noise. Robust feature extraction aims at finding new speech features that would be less dependent on the particular acoustic environment. They are usually motivated by some aspects of human speech

perception, due to humans' exceptional ability to recognize speech in noisy environments.

This paper starts by reviewing the main concepts of human speech perception in Section 2. The conventional features based on short-term power spectral estimates are reviewed in Section 3. This is followed by a description of several alternative feature extraction methods in Sections 4 and 5. Finally, the main conclusions are given in Section 6.

2 Human Speech Perception

Humans' exceptional ability to recognize speech, even under adverse conditions, has motivated the use of human speech perception knowledge in ASR. Practically all speech feature extraction methods in use today incorporate some properties of human speech perception, ranging from simple psycho-acoustic concepts to simulating the processes in the human auditory system in great detail. This section summarizes the properties of human speech perception that are commonly used in ASR. It starts with an overview of the physiology of the human auditory system, followed by a description of the most important psycho-acoustic results. Good overviews of the topic can be found in [9, 8, 4, 31].

2.1 Physiology of Human Auditory System

Human speech perception consists of converting sound pressure waves into the corresponding linguistic messages. In the context of speech perception, the human auditory system can be divided into two main parts, the pre auditory-nerve part and the post auditory-nerve part. The first part transforms the sound pressure waves

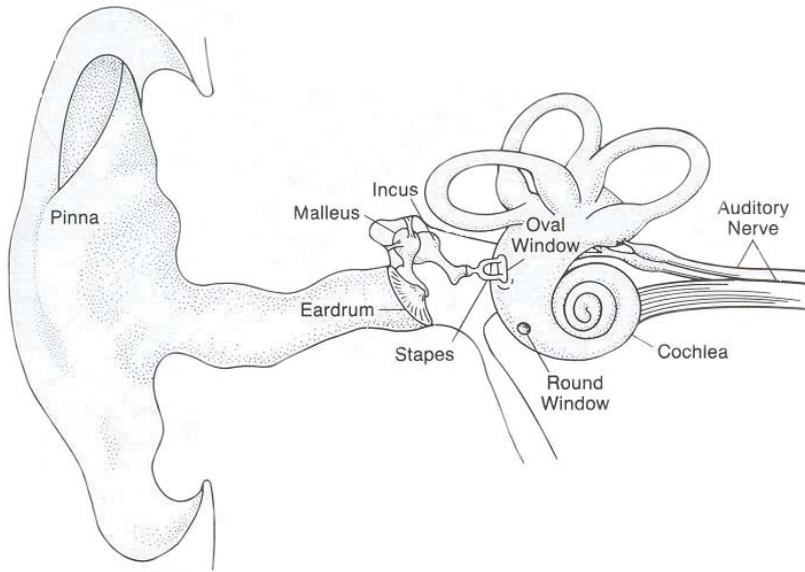


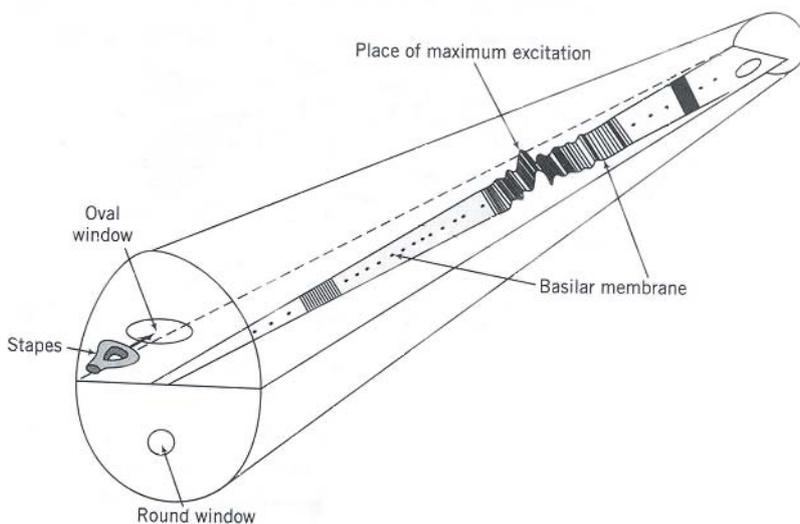
Figure 1 Pre auditory-nerve part of human auditory system [31]

into auditory-nerve activity, and is relatively well understood. The second part comprises higher level processing in the human brain, which transforms the auditory-nerve activity into a linguistic message. Little is known about this part of the system.

The pre auditory-nerve part of the human auditory system is illustrated in Figure 1. It consists of the outer ear, middle ear, and inner ear. The outer ear directs the sound pressure waves toward the eardrum. The middle ear converts the vibrations of the eardrum into vibrations at the oval window. It performs impedance matching between the air medium in the outer ear and fluid medium in the inner ear, and protects the inner ear from extensively intense sounds. Finally, the inner ear converts the mechanical vibrations at the oval window into electrical activity of auditory neurons.

The inner ear consists of the cochlea and the auditory nerve. Cochlea is a fluid-filled tube lon-

Figure 2 Simplified model of the cochlea [18]



gitudinally partitioned by the basilar membrane, as shown in Figure 2. The mechanical vibrations at the oval window excite the fluid inside the cochlea and cause the basilar membrane to vibrate. The displacement at a specific location along the basilar membrane is dependent on the frequency and the intensity of the input sound. Uniformly distributed along the basilar membrane are sensors, the inner hair cells, that transform the displacements of the basilar membrane into firings of the auditory neurons. It is said that a neuron fires when it exhibits an impulse in its electrical potential. The neural activity is processed further by the post auditory-nerve part of the human auditory system.

The frequency response of the basilar membrane varies along its length. The positions closest to the cochlea input are most sensitive to high frequencies, while those close to the apex are most sensitive to the low frequencies. In addition, frequency resolution of human hearing is highest in the low-frequency region, and it decreases gradually toward higher frequencies. Thus, the cochlea can be modeled by a bandpass filter bank, such that each filter models basilar membrane frequency response at a certain position. The logarithm of filter center frequencies is approximately proportional to the corresponding distance from the apex of the basilar membrane. Furthermore, filter bandwidths are proportional to the corresponding center frequencies, causing high frequency resolution at low frequencies, and vice versa. A filter bank specially designed to model the frequency response along the basilar membrane in great detail is usually referred to as a cochlear filter bank.

Neural activity generally increases with increased sound intensity due to the increased amplitude of the basilar membrane vibration. The extent of neural activity can be modeled as the logarithm of sound intensity. Furthermore, for frequencies up to 4 kHz, neural firings tend to be time-synchronized with the displacements of the basilar membrane in one direction. Experiments from auditory physiology have demonstrated that higher auditory processing centers in the brain make use of the synchrony information, as well as the average neural firing rate in the given channel.

2.2 Some Important Results from Psycho-Acoustics

Psycho-acoustics is the study of human auditory perception that relates acoustic signals to what the human listener perceives. Results from psycho-acoustics help distinguish the properties of speech signals that are important for human perception from those that are irrelevant. This section summarizes several psycho-acoustic results that have been used successfully in ASR.

2.2.1 Loudness as a Function of Sound Intensity and Frequency

Sound intensity is measured in terms of sound pressure level relative to a well-defined reference level, and is expressed in decibels. Perceived loudness is related to sound intensity, and is usually approximated as the logarithm of speech signal power. Alternatively, it can be approximated as the cubic root of signal power [37, 1].

Perceived loudness depends also on frequency. The sensitivity of human hearing is gradually reduced for frequencies lower than approximately 400 Hz and greater than 5 kHz [31].

2.2.2 Masking, Critical Bands and Bark Scale

Masking is an important phenomenon in hearing that denotes the fact that a tone (probe) that is clearly perceivable when presented in isolation, becomes imperceivable when presented together with another tone (masker). Consequently, the intensity of the probe has to be raised above the hearing threshold by a certain amount, called amount of masking, in order to be heard. The amount of masking increases with increased masker intensity, and with reduced difference between probe and masker frequencies.

Many masking phenomena can be explained using the notion of critical bands [31]. For example, a band of noise kept at constant intensity while its bandwidth is increased is perceived to have constant loudness until the critical bandwidth is reached. Thereafter, the loudness increases. Furthermore, when two sounds have energies inside the same critical band, the sound with higher energy inside the band dominates the perception and masks the other sound. Critical bandwidths are commonly approximated by the following expression [40]

$$CB = 25 + 75 [1 + 1.4 (F / 1000)^2]^{0.69}, \quad (1)$$

where CB is critical bandwidth and F is frequency, both given in Hertz.

Bark scale is a perceptually-warped frequency scale designed such that critical bandwidths have a constant value of 1 Bark along the entire scale. The mapping from the linear frequency scale to Bark scale is commonly approximated by the following expression [40]

$$F_{Bark} = 13 \arctan (0.76 F / 1000) + 3.5 \arctan (F / 500)^2, \quad (2)$$

where F is frequency given in Hertz, and F_{Bark} is the corresponding perceptual frequency given in

Bark. An older approximation often found in literature is given by [23]

$$F_{Bark} = 6 \ln \left(F/600 + \sqrt{(F/600)^2 + 1} \right). \quad (3)$$

Critical bandwidths correspond approximately to 1.5 mm spacing along the basilar membrane. Since the typical length of the basilar membrane is 35 mm, it is usually modeled by a set of 24 critical-bandwidth filters uniformly distributed along the Bark scale. Such a filter bank is referred to as critical-band filter bank, and is commonly used in speech feature extraction. Note that it is similar to the cochlear filter bank introduced in Section 2.1, but the motivation for its use comes from psycho-acoustics rather than auditory physiology.

2.2.3 Frequency Perception and Mel Scale

Probably the most commonly used perceptually-warped frequency scale, the mel scale, evolved from a set of experiments on human frequency perception [38]. The perceived frequency (pitch) of a 1 kHz tone at 40 dB sound pressure was defined as a reference point and assigned the value of 1000 mel. Listeners were then asked to adjust the tone frequency until the pitch they perceived was twice the reference, half the reference, and so on. The obtained frequencies were labeled 2000 mel, 500 mel, etc. In this way, the mapping between linear and mel frequency scales was found to be approximately linear for frequencies up to 1000 Hz and logarithmic for frequencies above 1000 Hz.

A commonly used analytic approximation [31] of the mapping is given by

$$F_{mel} = 2595 \log_{10}(1 + F / 700), \quad (4)$$

where F is the frequency in Hertz, and F_{mel} is the perceived frequency in mel. The relationship between mel and Bark scales is approximately linear. Thus, it is of little importance which of the scales is used in a practical application.

3 Features Based on Short-Term Power Spectrum of Speech

All commonly used speech features today are based on short-term power spectrum estimates of speech. The power spectrum estimates are usually modified to account for some psycho-acoustical concepts described in Section 2.2, e.g. perceptual-warping of frequency axis, and non-linear compression to account for the intensity-to-loudness conversion.

Furthermore, it is common to perform discrete cosine transform (DCT) on the spectral represen-

tation. This is done mainly because of the good decorrelation properties of DCT. Decorrelated feature vectors are desirable, since their statistical properties can be modeled by diagonal covariance matrices, which considerably reduces the computation complexity during both system training and recognition.

The parameters obtained by logarithmic compression of the power spectrum estimate followed by DCT are known as cepstral coefficients. Speech feature vectors usually consist of the cepstral coefficients and their first and second derivatives, in order to account for temporal changes in the speech spectrum.

Feature extraction methods differ in the spectral estimation methods used, and in the way the psycho-acoustical concepts are employed. This section describes some commonly used features based on short-term power spectrum estimates. A good overview of the spectral analysis techniques used in ASR can be found in [33].

3.1 Features Based on Linear Prediction Analysis

One conventional method for estimating short-term power spectrum of speech is based on the assumption that speech can be modeled as an autoregressive (AR) process of order p . Computation of the parameters of the AR model is equivalent to computation of the linear prediction (LP) coefficients of order p . For an analysis frame of length N , approximately $p^2 + N(p + 1)$ operations are required for computation of the LP coefficients. Typical values of predictor order used in ASR are between 8 and 14.

If the prediction order is properly chosen, LP coefficients provide a compact and precise representation of the speech spectral envelope, particularly for vowel-like sounds, which are completely characterized by their spectral peaks. However, the quality of the speech representation is highly dependent on the proper choice of the model order. If the model order is too low, not all of the prominent spectral peaks will be

properly modeled. If it is too high, the LP model tends to match random variations in the speech spectrum. Furthermore, the AR model is not well suited for modeling nasal sounds, which are characterized by spectral zeros. Finally, the major problem of the LP analysis is its sensitivity to noise.

Several different representations derived from the LP coefficients have been used in speech recognition including reflection coefficients, line-spectrum pair parameters, perceptually warped LP coefficients and LP-based cepstral coefficients [7]. Among them, linear prediction cepstral coefficients (LPCC) have been most successful, and are the only LP-based features used today. Detailed description of the use of LP in speech processing can be found in [29, 30, 34, 7].

3.2 Features Based on Subband Power Estimates

The short-term power spectrum can alternatively be estimated by a set of subband power estimates. They are computed by filtering a speech frame through a bandpass filter bank, and estimating the power of each subband signal. Cepstral coefficients are normally derived from the set of subband power estimates.

Perceptual warping of the frequency axis is easily implemented by choosing a filter bank with filters uniformly distributed along a perceptually warped frequency scale (e.g. mel or Bark scale).

Bandpass filtering and subband power estimation can be done both in time and frequency domain, as shown in Figures 3 and 4, respectively.

The advantage of the time-domain processing is the possibility for obtaining better frequency resolution at low frequencies and better time resolution at high frequencies, which is in agreement with human hearing. This is achieved by choosing shorter filter impulse responses at high frequencies than at low frequencies. On the other

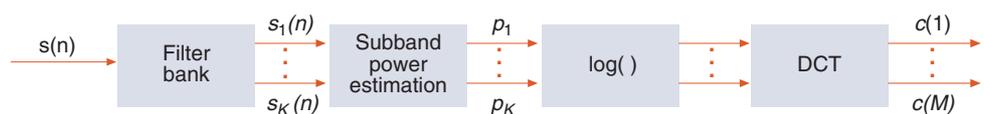


Figure 3 Time-domain computation of cepstral coefficients

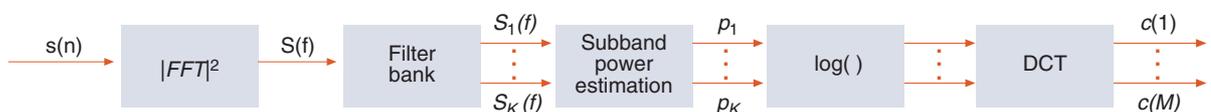


Figure 4 Frequency-domain computation of cepstral coefficients

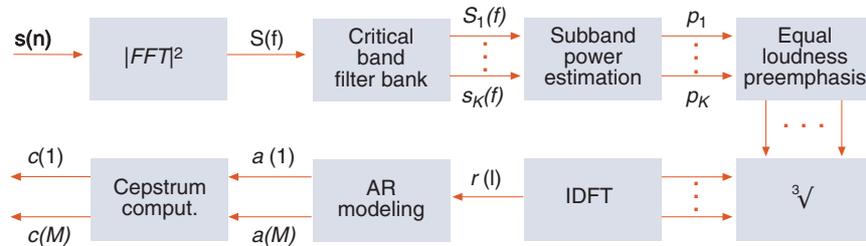


Figure 5 PLP method for speech feature extraction

hand, frequency domain processing is more computationally efficient. For the typical choice of parameters, the computation complexity of the frequency-domain processing is in the order of 10 operations per sample, similarly to the LP analysis, while the computation complexity of the time domain processing is in the order of 10^3 operations per sample.

Mel-frequency cepstrum coefficients (MFCC) [5] computed in the frequency domain according to Figure 4, are the most frequently used feature vectors in the state-of-the-art ASR systems. The filter bank consists of overlapping filters with triangular frequency responses uniformly distributed on the mel scale. Typically, 24 filters are used in the frequency range up to 8 kHz.

The popularity of the MFCC is due to their superior noise robustness compared to LP-based features [21], the simplicity with which perceptual warping is employed, and the low computational cost.

3.3 Perceptual Linear Predictive Analysis

Perceptual linear predictive (PLP) analysis [20, 22, 24, 23], illustrated in Figure 5, differs from the MFCC computation in two main aspects:

- Psycho-acoustic concepts are more accurately modeled;
- The perceptually-modified spectrum is fitted by an AR model.

As in MFCC computation, the analysis starts with an FFT-based power spectrum estimation, followed by critical-band filtering and subband power estimation. A minor difference compared to the MFCC computation is in the shape of filter frequency responses, which are chosen to better match the shape of the masking curves. In addition, the Bark scale is used instead of the mel scale.

The processing continues by equal-loudness preemphasis of the resulting spectral estimate. This is done in order to compensate for reduced sensitivity of the ear at low and high ends of the speech frequency range, as explained in Section

2.2.1. Next, cubic-root power compression is applied to the modified spectrum in order to model the intensity-to-loudness conversion. Finally, the perceptually modified spectral estimate is fitted by an AR model. This results in a set of LP-parameters that are usually converted into cepstral coefficients. Note that the AR modeling gives more weight to the high-energy parts of the spectrum than to the low-energy parts.

Seventeen critical-band filters are typically used to cover the frequency range up to 5000 Hz. The center frequencies are uniformly spaced on the Bark scale. The order of the AR model is typically between five and eight.

An advantage of PLP compared to the conventional LP analysis is the use of critical-band filtering and perceptual warping prior to AR modeling. The filtering provides spectral smoothing, which reduces the influence of the irrelevant spectral fine structure on the AR model. The perceptual warping reduces the weight given to the model fit at higher frequencies. This is in accordance with the reduced spectral resolution of human hearing at higher frequencies.

PLP was shown to outperform the conventional LP analysis in the ASR context both in clean speech and in the presence of additive noise. The computational complexity of the two methods is approximately the same.

4 Auditory Models and Their Modifications

Auditory models represent a class of speech representations that are based on simulating physiological processes in the human auditory system in great detail. Since human speech recognition is extremely robust, it is believed that the use of auditory models in ASR would lead to improved recognition performance in adverse conditions. However, since very little is known about human speech feature extraction beyond the auditory nerve level, auditory models include a considerable amount of heuristics.

This section describes two different auditory models and their modifications. Several other auditory models are described in [19].

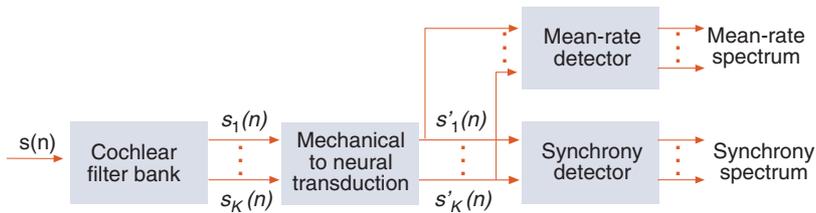


Figure 6 Joint synchrony/mean-rate auditory model

4.1 Joint Synchrony/ Mean-Rate Auditory Model

Joint synchrony/mean-rate auditory model [36] consists of three processing stages, as illustrated in Figure 6. In the first stage, the input signal is divided into a number of overlapping frequency bands using a cochlear filter bank. Forty band-pass filters with bandwidths 0.5 Bark were used, covering the frequency range between 130 Hz and 6400 Hz. This stage models the processes on the basilar membrane. The second stage consists of several nonlinearities that model the transformation from the basilar membrane vibrations to the probability of neural firings. The last stage consists of two branches. The first one consists of low-pass filtering of the output from the second stage in order to estimate the average rate of neural firings for each channel. The result is referred to as mean-rate spectrum. The second one uses the set of generalized synchrony detectors (GSD) to measure the extent of dominance of periodicities at subband center frequencies. The result is referred to as synchrony spectrum.

A simplified structure of a GSD is illustrated in Figure 7. It has two inputs, the output signal from the second stage and its delayed version, with the delay equal to the inverse of the subband center frequency, F_{c_k} . It computes the envelopes of the sum and difference of the two input signals, and outputs the ratio between the two envelopes. The synchrony spectrum consists of the outputs from GSDs for all channels.

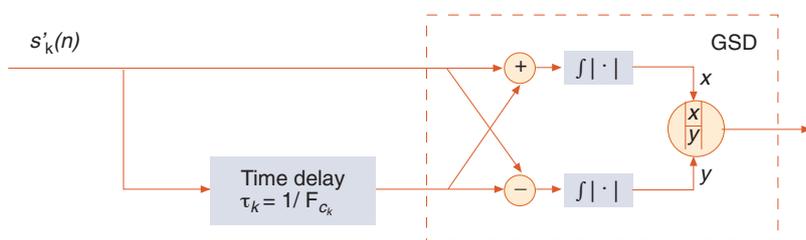


Figure 7 Simplified structure of generalized synchrony detector

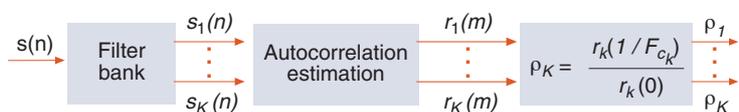


Figure 8 Time-domain based SBCOR analysis

An important property of the synchrony spectrum is that it enhances spectral peaks. If a subband signal has a dominant periodicity close to the subband center frequency, the difference between the two input signals to the GSD becomes very small, and the output from the GSD attains a large value. Thus, the output from the channel with the center frequency closest to a spectral peak position has a considerably larger value than the outputs from the neighboring channels.

An improved synchrony detector, referred to as average localized synchrony detector, has recently been proposed [3]. Preliminary studies indicate that it is capable of overcoming some shortcomings of the GSD.

Both mean-rate spectrum and synchrony spectrum have been used in ASR. The mean-rate spectrum is most suitable for determining acoustic boundaries and broad acoustic class classification. The synchrony spectrum, on the other hand, shows narrow peaks at the formant frequencies, and is suitable for more accurate phoneme classification.

4.2 Subband-Autocorrelation Analysis

Subband autocorrelation (SBCOR) analysis [25, 26], is a simplification of the synchrony spectrum computation described in Section 4.1. The main idea of measuring the dominance of periodicities at subband center frequencies remains the same, but the particular choice of the dominance measure is different. Furthermore, the computation is simplified by skipping the intermediate stage that simulates the mechanical-to-neural transduction in the inner ear.

SBCOR analysis is illustrated in Figure 8. It starts by passing a speech frame through a bandpass filter bank with center frequencies $\{F_{c_k}\}_{k=1}^K$. Then, for each subband signal the autocorrelation coefficient is computed at the time equal to the inverse of the center frequency, i.e.

$$\rho_k(\tau_k) = \frac{r_k(\tau_k)}{r_k(0)}, \text{ for } \tau_k = 1/F_{c_k}, \quad (5)$$

where $r_k(\cdot)$ denotes the autocorrelation function of the k -th subband signal. The resulting speech representation is referred to as SBCOR spectrum.

Alternatively, SBCOR analysis can be done in the spectral domain, as shown in Figure 9. In this case, an FFT-based spectral estimate is computed first, followed by subband filtering in the spectral domain. Finally, subband autocorrelation functions are found as inverse DFT of the

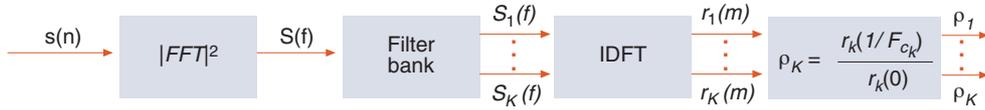


Figure 9 Frequency-domain based SBCOR analysis

subband spectra, and the autocorrelation coefficients are computed.

Generally, a spectral peak at frequency F gives rise to peaks in the autocorrelation function at times $\tau = n / F$, where n is an integer. Consequently, the value of subband autocorrelation coefficient at time $1 / F_{c_k}$ indicates the extent of dominance of the subband center frequency in the subband signal. Thus, the SBCOR spectrum provides a good indication of the positions of speech spectral peaks.

Several different filter banks for use with SBCOR analysis were compared in [25]. It was found that a fixed Q filter bank (i.e. fixed ratio between bandwidth and center frequency) with center frequencies uniformly spaced on the Bark scale gave the best results. Furthermore, it is not crucial whether the filter shape is similar to the cochlear filter or not. Both 128 and 16 filters were used in the recognition experiments with SBCOR analysis, but no attempt to optimize the number of filters has been reported.

SBCOR analysis was shown to outperform conventional speech feature extraction methods based on subband power estimates [25]. In addition, robustness of SBCOR spectrum against different types of speech distortion was demonstrated in [26].

4.3 Ensemble Interval Histograms

Ensemble Interval Histogram (EIH) [15, 17] is probably the best known auditory model used in ASR. It is based on temporal information in simulated neural firing patterns, similarly to the synchrony spectrum described in Section 4.1. However, the two auditory models differ in the way neural firing patterns are computed, and in the way temporal information is extracted.

4.3.1 Algorithm Description

The procedure for EIH computation is illustrated in Figure 10. A speech frame is passed through a bandpass filter bank that simulates the frequency response of the basilar membrane. The resulting subband signals model vibrations at different locations along the basilar membrane.

Transduction between basilar membrane vibrations and neural firings is modeled by an array of level-crossing detectors. The level values associated to the detectors correspond to different neural fiber firing thresholds.

Neural firings are simulated as positive-going level crossings. Temporal information is extracted from the neural firing patterns by measuring the inverse interval lengths between successive positive crossings of the same level, i.e.

$$f_{kl}(i) = \frac{1}{n_{kl}(i+1) - n_{kl}(i)}, \quad (6)$$

where $n_{kl}(i)$ denotes the location of the i -th positive-going crossing of the l -th level for the k -th subband signal.

Next, the frequency axis is divided into a number of histogram bins, R_j , and a histogram of the inverse interval lengths is constructed. The count of the j -th histogram bin is computed as

$$\text{count}(j) = \sum_k \sum_l \sum_i \Psi_j\{f_{kl}(i)\}, \quad (7)$$

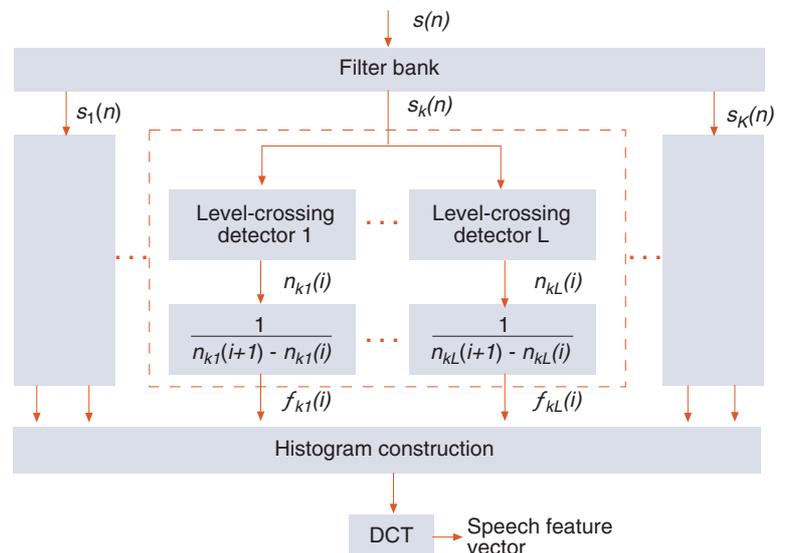
where

$$\Psi_j\{f_{kl}(i)\} = \begin{cases} 1 & f_{kl}(i) \in R_j \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The histogram is usually normalized by the sum of all histogram bin counts, and a DCT is performed for decorrelation purposes.

The computational cost of the EIH computation is very high due to the time-domain filtering, as well as the need for heavy oversampling of the high-frequency subband signals in order to increase accuracy of measured level-crossing

Figure 10 EIH method for speech feature extraction



locations. It was shown in [21] that the EIH computation was three times slower than that of the synchrony spectrum, and as much as 360 times slower than MFCC computation.

The inverse level-crossing intervals are closely related to instantaneous dominant subband frequencies. Thus, histogram bins having large counts indicate dominant frequency regions, and EIH can be seen as an alternative spectral representation of speech that emphasizes spectral peaks.

4.3.2 Parameter Choice

ASR performance of EIH features depends on the choice of the analysis frame lengths, filter bank, number and location of the levels, and particular histogram bin allocation.

Analysis frame lengths that are inversely proportional to subband center frequencies have been used in order to ensure that analysis frames for different subband signals incorporate approximately the same number of signal periods (e.g. 10 or 20).

The filter bank typically consists of 85 bandpass filters uniformly distributed on a perceptually based scale in the range [0,4000 Hz]. Cochlear filters were originally used, but it was later shown that the particular filter shape was not important for good ASR performance [14].

Five level-crossing detectors have been used, with levels uniformly distributed on the logarithmic scale. However, the optimal choice of the levels is dependent on the signal intensity, and there is no well defined procedure for optimal level determination. It was shown in [28] that the performance of EIH is highly dependent on the choice of number of levels and their values.

Two different bin allocation schemes were compared in [17]. In the first one, 128 bins were uniformly distributed on the linear frequency scale, while in the second one 32 bins were uniformly distributed on a perceptually-based frequency scale. They led to approximately the same ASR performance.

4.3.3 Experimental Results

The performance of EIH has been compared to that of the conventional methods on several ASR tasks [15, 16, 21, 35]. A general conclusion is that EIH performs better than conventional methods in noisy conditions, while it performs slightly worse in clean conditions. However, the addition of delta and delta-delta parameters leads to smaller improvement in performance than for the conventional methods [35, 28]. EIH has also been shown to outperform both PLP and SBCOR methods on a small vocabulary ASR task in the presence of different types additive noise [28].

However, EIH performance in noisy conditions is still considerably lower than that of human listeners [17].

4.4 Zero Crossings with Peak Amplitudes

Kim et al. [28] derived an analytic expression for the variance of level-crossing interval lengths of a sinusoidal signal in the presence of white Gaussian noise. It showed that the variance increases with increased level value. Consequently, lower level values provide more reliable level-crossing interval lengths in the presence of noise. On the other hand, experimental results on an ASR task revealed the importance of the intensity information provided by properly chosen higher level values.

Motivated by these results a modification of the EIH method was proposed [28]. The set of level-crossing detectors was exchanged by a single zero-crossing detector, while intensity information was preserved by measuring peak amplitudes between successive zero crossings. Thus, the resulting speech parameterization, referred to as zero crossings with peak amplitudes (ZCPA), provides more reliable interval lengths in noisy conditions without sacrificing the intensity information. In addition, it circumvents the problem of choosing proper level values, and reduces the computational cost compared to the EIH method.

4.4.1 Algorithm Description

The procedure for ZCPA computation is illustrated in Figure 11. The input speech frame is passed through a bandpass filter bank. Each subband signal, $s_k(n)$, is then processed by a zero-crossing detector in order to determine the positions of all positive-going zero crossings. Then, for each pair of successive zero crossings, $z_k(i)$ and $z_k(i+1)$, the peak signal value, $p_k(i)$, and the inverse zero-crossing interval length, $f_k(i)$, are computed as

$$p_k(i) = \max_{z_k(i) \leq n < z_k(i+1)} \{s_k(n)\} \quad (9)$$

$$f_k(i) = \frac{1}{z_k(i+1) - z_k(i)}. \quad (10)$$

Next, the frequency axis is divided into a number of histogram bins, R_j , and a histogram of the inverse zero-crossing interval lengths is collected over all subband signals. However, instead of increasing the bin counts by one, they are increased by the logarithm of the corresponding signal peak amplitude. Thus, the count of j -th histogram bin is computed as

$$\text{count}(j) = \sum_k \sum_i \Psi_j \{f_k(i)\}, \quad (11)$$

where

$$\Psi_j\{f_k(i)\} = \begin{cases} \ln(p_k(i)) & f_k(i) \in R_j \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

Finally, DCT is performed on the histogram for decorrelation purposes.

The computational cost of the ZCPA method is lower than that of the EIH method. However, with the typical choice of parameters it is still two orders of magnitude higher than that of the standard MFCC method [10, 13]. This is due to the use of time-domain filtering and a need for heavy interpolation of high-frequency subband signals in order to reliably determine zero-crossing positions. The interpolation provides a larger number of points between subsequent zero crossings, and thus better frequency resolution. The computational cost depends mainly on the number and order of subband filters, required interpolation factors, and the order of the interpolation filter.

4.4.2 Relationship to Spectral Analysis

The ZCPA feature extraction method was derived from the EIH method, which was motivated by physiological processes in the human auditory system. However, from the signal processing point of view, ZCPA histograms can be seen as alternative short-term spectral representations of speech. This is explained in the following.

The dominant frequency principle [27] states that if there is a significantly dominant frequency in the signal spectrum, then the inverse zero-crossing interval lengths tend to take values in the vicinity of the dominant frequency. Thus, the inverse zero-crossing interval lengths, $f_k(i)$, of the k -th subband signal can be seen as estimates of the dominant subband frequency. Furthermore, the peak signal value between subsequent zero crossings, $p_k(i)$, can be seen as a measure of the instantaneous power in the subband signal. Consequently, the construction of ZCPA histograms consists of assigning subband power estimates to frequency bins corresponding to dominant subband frequencies. The standard MFCC method, on the other hand, assigns subband power estimates to entire subbands, without taking into account the power distribution within subbands. Thus, the ZCPA representation can be seen as an alternative spectral representation of speech that emphasizes spectral peaks, while de-emphasizing the information in spectral valleys, which is usually corrupted by noise.

4.4.3 Parameter Choice

The ZCPA representation depends on the choice of analysis frame lengths, the filter bank, and

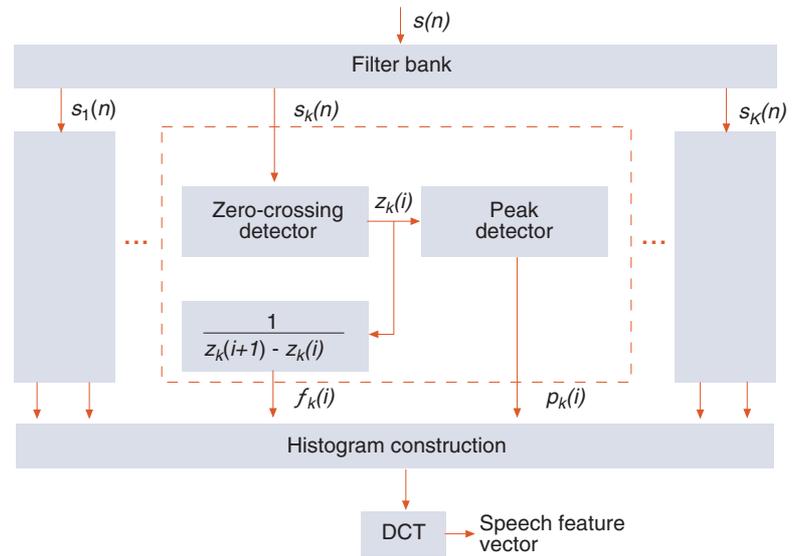


Figure 11 ZCPA method for speech feature extraction

histogram bin allocation. An attempt to optimize the parameter values with respect to ASR performance was made in [10, 13]. It was shown that a proper choice of analysis frame lengths and filter bandwidths was important for good recognition performance, while the performance was not very sensitive to the choice of the number of filters and histogram bins.

The best performance in the presence of noise was achieved using frequency dependent analysis frame lengths that ranged from 33 ms to 134 ms, 16 overlapping bandpass filters with bandwidths equal to 2 Bark, and 60 histogram bins. Both filters and histogram bins were uniformly distributed on the Bark scale in the range [0,4000 Hz].

4.4.4 Experimental Results

In an experimental study on a small-vocabulary isolated-word task [28], ZCPA features were shown to greatly outperform LPCC, MFCC, SBCOR, PLP and EIH features in the presence of additive background noise.

In another study [13, 10], the greater robustness of the ZCPA features compared to MFCC was confirmed on both a small-vocabulary isolated-word task and a medium-vocabulary continuous-speech task in several different environmental conditions. The advantage of using the ZCPA features was largest on the small-vocabulary task and in the presence of white noise. On the other hand, MFCC features performed better than ZCPA features on the clean speech. The results are illustrated in Figures 13 and 14 in the next section.

Furthermore, it was shown that the superior robustness of the ZCPA is partly due to the use of dominant frequency information and partly to

the time-domain processing. The improvement due to the use of dominant frequency information was considerable on both recognition tasks, and was largest in the presence of noise types with relatively flat spectral characteristics. The improvement due to the time-domain processing was considerable only on the small-vocabulary task.

5 Features Based on Subband Spectral Centroids

The presence of additive background noise results in changes of the power spectrum of the speech signal. However, the spectral peak positions remain practically unaffected, as long as the noise is added at moderate levels and does not have strong spectral peaks.

Auditory models and their modifications presented in Section 4 all utilized the information about dominant periodicities in the speech signal. This might be a major reason for their better robustness against additive background noise compared to the conventional speech features.

Reliable estimation of spectral peak positions is a difficult task, especially in the presence of noise. Thus, rather than estimating spectral peak positions directly, Paliwal [32] was concerned with deriving features for use in ASR that would convey the information about spectral peak positions. He studied the properties of subband spectral centroids (SSC) that are computed as the first moment of speech subband power spectra, i.e.

$$C_k = \frac{\int f S_k^\gamma(f) df}{\int S_k^\gamma(f) df}, \quad (13)$$

where $S_k(f)$ is the power spectrum of the k -th subband signal, and γ is a constant controlling the spectral dynamic range. Note that SSC depend on the particular filter bank and power spectrum estimate.

It was shown in [32] that SSC are closely related to spectral peak positions, and their robustness to additive white Gaussian noise was demonstrated.

Several experimental studies [32, 39, 2, 6, 10] have been performed where standard feature vectors were augmented by several SSC. However, the reported results on the effect of the added SSC on the recognition performance are not consistent. Thus, a better method of integrating SSC into speech feature vectors had to be found. One such method is described in the following.

5.1 Subband Spectral Centroid Histograms

The robustness of ZCPA features against additive background noise has indicated the positive effect of integrating dominant subband frequency information and subband power information into speech feature vectors. However, the high computational cost of the ZCPA method makes this method less attractive in practical applications.

On the other hand, it was shown that SSC can serve as reasonable estimates of dominant frequencies both for clean and noisy speech. Furthermore, they can be computed efficiently from a short-term speech power spectrum estimate.

Thus, the idea was to construct a feature extraction method that would combine the dominant-frequency information provided by SSC with subband power information in a similar way as in the ZCPA method. The resulting features are referred to as Subband Spectral Centroid Histograms (SSCH) [10, 11, 12].

5.1.1 Algorithm Description

The procedure for SSCH computation is illustrated in Figure 12. It starts by estimating the power spectrum of the given speech frame, followed by bandpass filtering in the frequency domain. Next, subband spectral centroids are estimated from the samples of the subband power spectra as

$$C_k = \frac{\sum_i i S_k^\gamma(i)}{\sum_i S_k^\gamma(i)}, \quad (14)$$

and subband power estimates are computed by integrating the subband power spectra over each subband. Alternatively, the integration can be done over a smaller frequency range centered around the subband spectral centroid. This might provide more robust estimates since the frequency area around the dominant frequency is less influenced by noise than the other parts of the subband. On the other hand, smaller integration areas lead to less reliable estimates.

The speech frequency range is then divided into a number of histogram bins, R_j , and a histogram of subband spectral centroids is constructed. The bin counts are increased by the logarithm of the corresponding subband power estimate normalized by the subband bandwidth. The normalization is done in order to avoid biasing of the histograms toward higher frequencies due to increased filter bandwidths. Thus, the count of the j -th histogram bin is computed as

$$\text{count}(j) = \sum_k \Psi_j \{C_k\}, \quad (15)$$

where

$$P_j \{C_k\} = \begin{cases} \ln(p_k / N_k) & C_k \in R_j \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

where N_k is the number of frequency samples in the k -th subband, that is proportional to the subband bandwidth. Finally, the DCT of the histogram is computed for decorrelation purposes.

The computational cost of the SSCH method is of the same order of magnitude as that of the MFCC method.

5.1.2 Relationship to MFCC

The SSCH and MFCC feature extraction methods have several common processing steps, i.e. spectral estimation, bandpass filtering and subband power computation. However, the SSCH method incorporates two additional steps, namely, SSC computations and SSC histogram construction.

While the MFCC method assigns a subband power to the entire frequency subband, in the SSCH method it is assigned to the histogram bin that contains the dominant subband frequency. In this way, the locations of spectral peaks are much better preserved.

Thus, similarly as the ZCPA method, the SSCH method can be seen as an alternative way of performing spectral analysis, which emphasizes spectral peaks. However, it is important to remember that SSC are only estimates of dominant subband frequencies computed from the speech spectra. Thus, they are affected by noise even if the true spectral peaks remain unchanged.

5.1.3 Choice of parameters

An experimental study aimed at optimizing some of the parameters involved in SSCH computation is described in [10]. The main results are given in the following.

The dynamic range of the power spectrum used in the SSC computation is controlled by the parameter γ in Equation 14. If γ is too small (near 0), SSC would approach the centers of their subbands and would thus contain no information. If it is too large (near ∞), SSC would correspond to the locations of the subband peak values of the FFT-based power spectrum, and would thus be noisy estimates. In the experimental study the best overall performance across all signal-to-noise ratios (SNRs) was achieved for $\gamma = 1$. However, some improvement was observed at low SNRs for increased value of γ , but this led to corresponding performance degradation at higher SNRs.

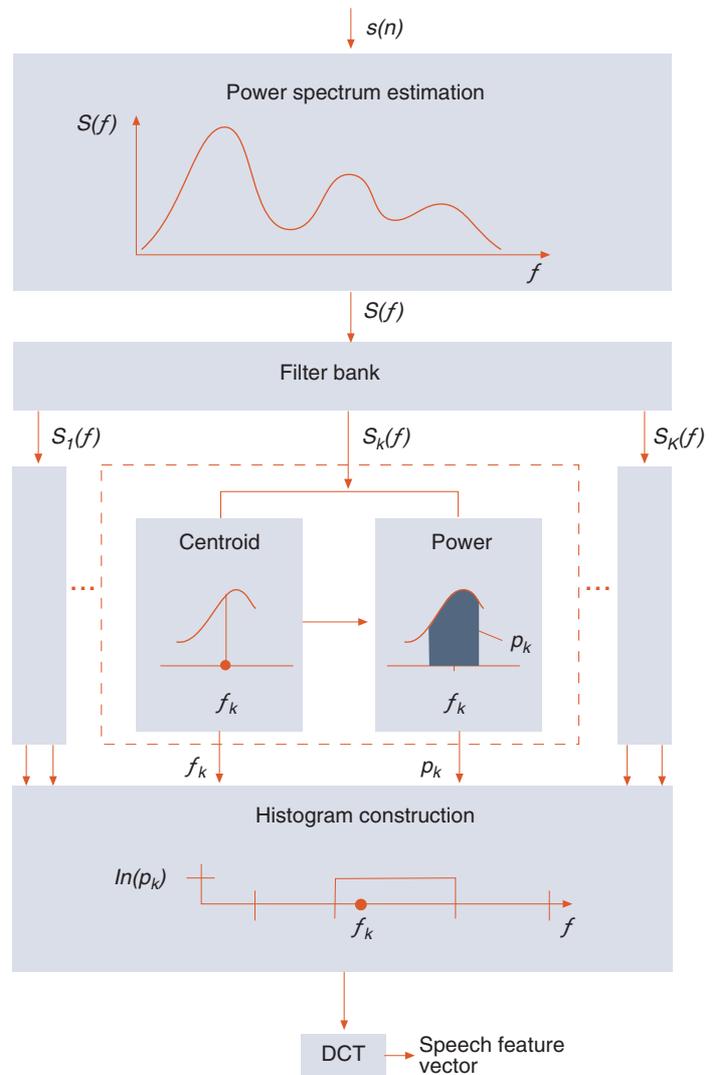


Figure 12 SSCH method for speech feature extraction

Filter bandwidths should ideally be chosen such that each subband contains exactly one dominant spectral peak. In this case, SSC would serve as good estimates of spectral peak positions. Too small filter bandwidths would result in a number of subbands that do not contain any dominant spectral peak. Centroids of such subbands would be sensitive to random variabilities in speech. On the other hand, if filter bandwidths stretch over several dominant spectral peaks, SSC would no longer represent reasonable estimates of subband peak locations. It was shown that the choice of filter bandwidths had a significant effect on the recognition performance, especially at low SNRs. The best results in the presence of noise were achieved using filter bandwidths equal to 3 Bark. On the other hand, the recognition performance was not sensitive to the particular number of filters.

Histogram bins should be sufficiently small to provide a good frequency resolution, but not too small to make the resulting speech parameterization sensitive to small fluctuations in spectral

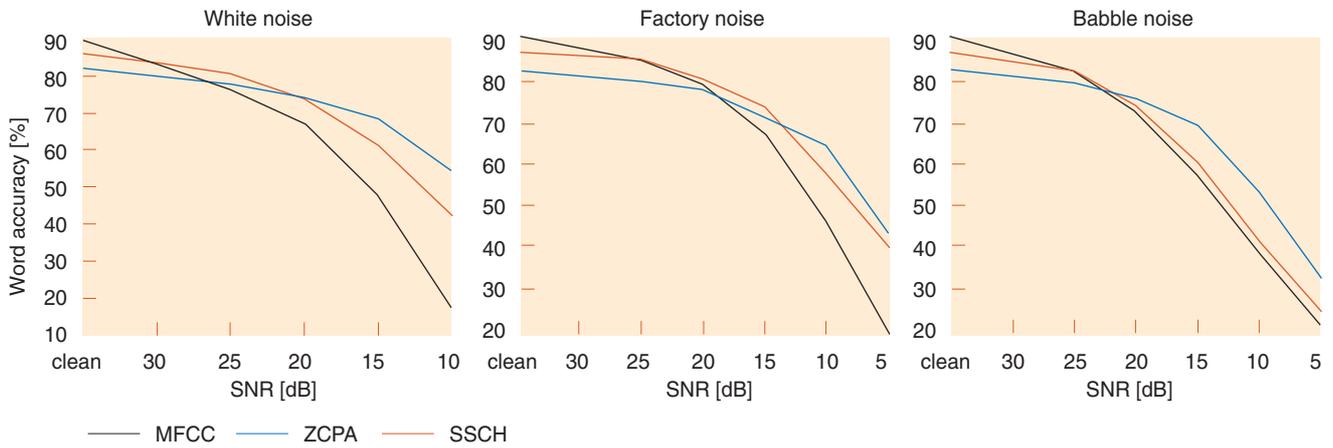


Figure 13 Comparison of MFCC, ZCPA and SSCH features on a small-vocabulary isolated-word recognition task

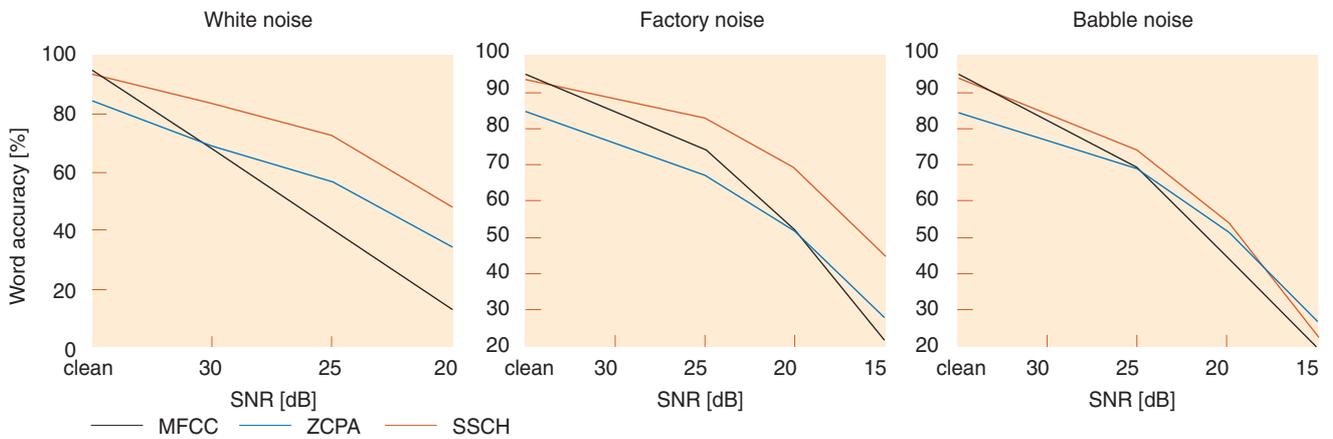


Figure 14 Comparison of MFCC, ZCPA and SSCH features on a medium-vocabulary continuous-speech recognition task

peak positions (e.g. due to speaker differences). It was shown that the recognition performance was not very sensitive to the particular number of histogram bins.

Finally, there was no significant performance improvement when subband power estimates were computed over a smaller frequency range around the subband centroid rather than over the entire subband.

5.1.4 Experimental Results

In the experimental study described in [10], the SSCH features were compared to both MFCC and ZCPA features on a small-vocabulary isolated-word recognition task and a medium-vocabulary continuous-speech task. Three different noise types were artificially added to the clean speech at several different SNRs. Note that the SNR was determined as the ratio between maximal speech frame power and the average noise power for the given speech file.

The SSC were computed from the FFT-based spectral estimates, using the dynamic range

parameter $\gamma = 1$. The filter bank consisted of 48 highly overlapping bandpass filters with rectangular frequency responses, covering the frequency range up to 4000 Hz. The filters were uniformly distributed on the Bark scale, with bandwidths equal to 3 Bark. The frequency range was divided into 38 histogram bins uniformly distributed on the Bark scale.

The recognition results on the two recognition tasks are shown in Figures 13 and 14. We can see that SSCH features consistently outperformed MFCC in the presence of additive noise on both recognition tasks. The advantage of using SSCH features was largest in the presence of white noise, while it was very small in the presence of background speech (i.e. babble noise). The advantage of using SSCH features compared to MFCC features was shown to be mainly due to the use of dominant subband frequency information in the SSCH method.

On the small-vocabulary task, SSCH features performed slightly better than ZCPA features at high SNRs, while ZCPA features performed

better at low SNRs. However, on the medium-vocabulary task, SSCH features performed considerably better than ZCPA features in practically all testing conditions. In addition, the computational complexity of the SSCH method is two orders of magnitude lower than that of the ZCPA method.

6 Conclusions

This paper has presented an overview of several alternative methods for speech feature extraction in ASR, which were motivated by the results from human speech perception. The methods have generally shown greater noise robustness compared to the conventional feature extraction methods.

The major reason for the greater robustness is probably the use of the information about dominant frequencies in the speech signal (i.e. spectral peak positions), as this information is not sensitive to the changes in environmental noise.

The dominant-frequency information is extracted in different ways in the different feature extraction methods, e.g. using the generalized synchrony detectors, subband autocorrelation coefficients, level-crossing statistics and subband spectral centroids. The best recognition results in the presence of environmental noise have been achieved by combining the subband power information, used by the conventional feature extraction methods, with the dominant subband frequency information. This can be done in a simple and computationally efficient way by computing the subband spectral centroid histograms.

References

- 1 Ainsworth, W A et al. Speech processing by man and machine. In: *Recognition of Complex Acoustic Signals*, Life Science Research Report 5, 307–351, 1977.
- 2 Albesano, D et al. A study of the effect of adding new dimensions to trajectories in the acoustic space. In: *Proc. Eur. Conf. on Speech Commun. and Technol. (EUROSPEECH)*, 4, Budapest, Hungary, 1503–1506, Sept. 1999.
- 3 Ali, A M A, der Spiegel, J V, Mueller, P. Robust auditory-based speech processing using the average localized synchrony detection. *IEEE Trans. Speech Audio Processing*, 10, 279–292, 2002.
- 4 Allen, J B. Cochlear modeling. *IEEE ASSP Mag.*, 2, 3–29, 1985.
- 5 Davis, S B, Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust., Speech, Signal Processing*, 28, 357–366, 1980.
- 6 De Mori, R et al. Ear-model derived features for automatic speech recognition. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, 3, Istanbul, Turkey, 1603–1606, 2000.
- 7 Deller, J R, Proakis, J G, Hansen, J H L. *Discrete-Time Processing of Speech Signals*. Upper Saddle River, NJ, Prentice Hall, 1993.
- 8 Fant, C G M. *Speech Sounds and Features*. Cambridge, MA, M.I.T. Press, 1973.
- 9 Flanagan, J. *Speech Analysis Synthesis and Perception, 2nd ed.* Springer-Verlag, 1972.
- 10 Gajic, B. *Feature Extraction for Automatic Speech Recognition in Noisy Acoustic Environments*. Trondheim, Norway, Norwegian University of Science and Technology, 2002. PhD thesis.
- 11 Gajic, B, Paliwal, K K. Robust feature extraction using subband spectral centroid histograms. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, 1, Salt Lake City, USA, 85–88, May 2001.
- 12 Gajic, B, Paliwal, K K. Robust parameters for speech recognition based on subband spectral centroid histograms. In: *Proc. Eur. Conf. on Speech Commun. and Technol. (EUROSPEECH)*, 1, Aalborg, Denmark, 591–594, Sept. 2001.
- 13 Gajic, B, Paliwal, K K. Robust speech recognition using features based on zero crossings with peak amplitudes. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Hong Kong, China, Apr. 2003.
- 14 Ghitza, O. Robustness against noise : The role of timing-synchrony measurement. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Dallas, USA, 2372–2375, Apr. 1987.
- 15 Ghitza, O. Temporal non-place information in the auditory-nerve firing patterns as a front-end for speech recognition in a noisy environment. *J. Phonetics*, 16, 55–76, 1988.
- 16 Ghitza, O. Auditory nerve representation as a basis for speech processing. In: *Advances in Speech Signal Processing*. Furui, S and

- Sondhi, M (eds.), 453–485. Marcel Dekker, Inc., 1992.
- 17 Ghitza, O. Auditory models and human performance in tasks related to speech coding and speech recognition. *IEEE Trans. Speech Audio Processing*, 2, 115–132, 1994.
 - 18 Gold, B, Morgan, N. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, 2000.
 - 19 Greenberg, S (ed.). *J. Phonetics*, 16, 1988. (Theme issue Representation of Speech in the Auditory Periphery)
 - 20 Hermansky, H. Perceptual linear predictive (PLP) analysis of speech. *J. Acoust. Soc. Am.*, 87, 1738–1752, 1990.
 - 21 Jankowski, C R, Vo, H-D H, Lippmann, R P. A comparison of signal processing front ends for automatic word recognition. *IEEE Trans. Speech Audio Processing*, 3, 286–293, 1995.
 - 22 Junqua, J-C. *Toward Robustness in Isolated-Word Automatic Speech Recognition*. Nancy, France, Univ. Nancy I, 1989. PhD thesis.
 - 23 Junqua, J-C, Haton, J-P. *Robustness in Automatic Speech Recognition – Fundamentals and Applications*. Kluwer Academic Publishers, 1996.
 - 24 Junqua, J-C, Wakita, H, Hermansky, H. Evaluation and optimization of perceptually-based ASR front-end. *IEEE Trans. Speech Audio Processing*, 1 (1), 39–48, 1993.
 - 25 Kajita, S, Itakura, F. Subband-autocorrelation analysis and its application for speech recognition. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, 2, 193–196, 1994.
 - 26 Kajita, S, Itakura, F. Robust speech feature extraction using SBCOR analysis. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, 1, Detroit, USA, 421–424, May 1995.
 - 27 Kedem, B. Spectral analysis and discrimination by zero-crossings. *Proc. IEEE*, 74, 1477–1493, Nov. 1986.
 - 28 Kim, D-S, Lee, S-Y, Kil, R M. Auditory processing of speech signals for robust speech recognition in real-world noisy environments. *IEEE Trans. Speech Audio Processing*, 7, 55–69, 1999.
 - 29 Makhoul, J. Linear prediction : A tutorial review. *Proc. IEEE*, 63, 561–580, Apr. 1975.
 - 30 Markel, J D, Gray, A H Jr. *Linear Prediction of Speech*. Springer-Verlag, 1976.
 - 31 O’Shaughnessy, D. *Speech Communication : Human and Machine*. Addison-Wesley, 1987.
 - 32 Paliwal, K K. Spectral subband centroid features for speech recognition. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, 2, Seattle, USA, 617–620, May 1998.
 - 33 Picone, J W. Signal modelling techniques in speech recognition. *Proc. IEEE*, 81, 1214–1247, Sept. 1993.
 - 34 Rabiner, L, Schafer, R W. *Digital Processing of Speech Signals*. Englewood Cliffs, NJ, Prentice Hall, 1978.
 - 35 Sandhu, S, Ghitza, O. A comparative study of mel cepstra and EIH for phone classification under adverse conditions. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Detroit, USA, 409–412, May 1995.
 - 36 Seneff, S. A joint synchrony/mean-rate model of auditory speech processing. *J. Phonetics*, 16, 55–76, Jan. 1988.
 - 37 Stevens, S S. On the psychophysical law. *Psychol. Rev.*, 64, 153–181, 1957.
 - 38 Stevens, S S, Volkman, J. The relation of pitch of frequency : A revised scale. *Am. J. Psychol.*, 53, 329–353, 1940.
 - 39 Tsuge, S, Fukada, T, Singer, H. Speaker normalized spectral subband parameters for noise robust speech recognition. In: *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Phoenix, USA, May 1999.
 - 40 Zwicker, E, Terhart, E. Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *J. Acoust. Soc. Am.*, 68, 1523–1525, 1980.

Adaptation Techniques in Automatic Speech Recognition

TOR ANDRÉ MYRVOLL



Tor André Myrvoll (33) received his *Siv.Ing. (MSc)* and *Dr.Ing. (PhD)* degrees from the Department of Telecommunications, Norwegian University of Science and Technology, in 1997 and 2002 respectively. From 2001 to 2002 he worked as a consultant at the Multimedia Communications Research Laboratory, Bell Laboratories in Murray Hill. He is now a researcher at the Brage project, funded by the Norwegian Research Council.

myrvoll@tele.ntnu.no

Automatic speech recognition (ASR) has come a long way in the last two decades, and is now on the verge of being a mature technology ready for a wide range of commercial uses. However, as ASR systems are being used in practical applications like dialog systems and dictation software, they are deployed outside of the controlled environment of the laboratories where they were built. It is a well known fact that a mismatch between the training data used to build the acoustic model, and the test conditions leads to a significant drop in recognition performance, maybe even rendering the system useless for some users. Such mismatches can be the result of adverse acoustic conditions, channel variations, different speaking styles or a combination of these factors. In this review article we will focus on the model adaptation approach as one way to alleviate these inevitable mismatches. In this approach the acoustic model is tuned to match its working environment, either in a supervised manner before deployment, or totally unsupervised, relying on the system to automatically match the different conditions it will encounter. The intention behind this review is not to mention every research direction, but rather to give a comprehensive treatment of a few topics that are of practical and theoretical interest, as well as pointing out some promising directions of future research.

1 Introduction

Today, after decades of research, more and more ASR systems are being deployed for everyday use. Although we are still short of having systems that perform satisfactorily for natural and spontaneous speech, the performance on some dictation tasks and constrained dialogs is good enough for many applications. Clearly, whenever a new ASR system is deployed in a practical working environment, we would like it to have the same performance regardless of who is using it, and under what conditions. This, unfortunately, is not the case in the real world. Whenever the ASR system encounters a working condition that is not consistent with the training condition, the performance will show a significant decrease. As an example, the Wall Street Journal dictation shows an absolute word error rate (WER) increase from 4 % to 30 %, when non-native American English speakers with detectable accents are encountered.

The goal of model adaptation is to find algorithms that automatically “tune” a given hidden Markov model (HMM) to a new test environment using limited, but representative new data. The tuning should ultimately result in a new, *adapted* HMM that improves ASR performance, i.e. giving us a lower word error rate. In turn, this reduced word error rate will lead to a direct improvement on dictation and transcription tasks, as well as improved semantic error rates for most dialog tasks. In the Wall Street Journal example given earlier, the use of model adaptation was able to reduce the WER from 30 % to 14 % using only 40 utterances [1]. Model adaptation is also useful when the amount of training data for a specific task is insufficient to train a model of a certain complexity. Instead of recording and transcribing more data, one can simply

adapt an existing model, if available, using the task dependent data.

In this review article we will try to communicate the underlying principles of model adaptation, as well as present the major adaptation scenarios, which are supervised (or batch) adaptation, and unsupervised (online) adaptation. We will also present what we take to be the major approaches to model adaptation today – maximum a posteriori (MAP) adaptation, transformation based adaptation and adaptive training. First we give a brief presentation of the de facto standard of acoustic modeling – the hidden Markov model.

2 Decision Theory and the HMM

An ASR system is in principle a statistical pattern recognizer that approximates the decision rule,

$$\hat{W} = \arg \max_W p(W|\mathbf{O}), \quad (1)$$

where $\mathbf{O} = [\mathbf{o}_1, \dots, \mathbf{o}_T]$ is a sequence of feature vectors representing the spoken utterance and W is a word string. The function $p(W|\mathbf{O})$ is the probability of a word string given the spoken utterance. Using Bayes theorem we can write the equivalent expression,

$$\hat{W} = \arg \max_W p(\mathbf{O}|W)P(W), \quad (2)$$

where we see that the posterior probability of the string, $p(W|\mathbf{O})$, has been replaced by the product of the likelihood of the feature vectors, \mathbf{O} , given the string, W , and the a priori probability of the string, W . These two factors are known as the acoustic- and language models, respectively.

The language model will not be mentioned further here, although the adaptation of language models is an important research subject [2].

We clearly do not know the true statistical nature of human speech, and so we approximate the true distribution, $p(\mathbf{O}|W)$, by a model, $p(\mathbf{O}|W, \Lambda)$, from a family \mathcal{F} of parametric distributions, Λ being the parameter vector specifying the actual distribution. In ASR the model of choice is the HMM, mainly because of its modeling flexibility, as well as the existence of effective algorithms for model training and likelihood evaluation. The model parameter vector, Λ , is usually found in a maximum likelihood sense, that is,

$$\Lambda = \arg \max_{\Lambda'} p(\text{Training data} | \Lambda'). \quad (3)$$

The resulting model is then plugged directly into the decision rule given in equation (2) instead of the exact acoustic model. This practice is referred to as the *plug-in MAP decision rule*.

A short description of the HMM now follows. The basic building block of the HMM is a *discrete Markov chain*. This is a random sequence of elements, q_1, q_2, \dots, q_T , which we will refer to as *states*, drawn from a finite set. Let us write the probability of the next state in the sequence as $p(q_t | q_{t-1}, q_{t-2}, \dots, q_1)$. If this probability is equal to $p(q_t | q_{t-1})$ for all t , we have a first order Markov chain. The use of Markov chains enables us to model the short time stationarity of the speech signal, as the system can stay in the same state for the duration of e.g. a vowel, and then jump into a new state as the next sound in the utterance is observed. Of course, in an HMM

the states are never observed directly, but rather as random observations distributed according to a state dependent probability density function (pdf), $p(\mathbf{o}_t | q_t = s)$. Here s is the state that is occupied at time t . These pdfs are typically in the form of a mixture of multivariate normal distributions,

$$p(\mathbf{o}_t | q_t = s) = \sum_{k=1}^K c_{sk} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{sk}, \boldsymbol{\Sigma}_{sk}), \quad (4)$$

where c_{sk} is the mixture weight and $\boldsymbol{\mu}_{sk}$ and $\boldsymbol{\Sigma}_{sk}$ are the mean vector and covariance matrix of state s , mixture K , respectively.

For adaptation purposes, the most important model parameters of the ones we have described until now are the mixture mean vectors. Although techniques exist to adapt all of the HMM parameters, this is rarely done in practice. Adapting the state transition probabilities and mixture weights rarely has any significant effect, and the covariance matrix should always be treated with the utmost care, as this is one of the hardest parameters to estimate robustly. With this in mind, the reader should be able to appreciate the one-sided focus on the mean vectors in the sections to come.

3 Model Adaptation Principles

During HMM training the model parameters are found by maximizing the likelihood of the training data. We want our adaptation algorithms to be consistent with this paradigm, but as adaptation data can be scarce, the direct use of ML estimators as in training may not be a feasible option, and so we will have to make some strong assumptions to make adaptation practical.

The main assumption is that HMMs trained in different application scenarios should contain much of the same information. For example, any HMM will implicitly contain knowledge about the relative positions of formants for different vowels. Assuming that these relative positions are similar from speaker to speaker, we could in principle adapt all the vowel models for a new speaker model using only a few utterances to determine the necessary frequency scaling. Although no explicit mapping exists that will transform model parameters to reflect a specific speaking apparatus, the principle is clear. We should expect the initial model to contain all the relevant information about the acoustic signal embedded in its structure. Any adaptation to a new domain should not try to re-learn this information, but instead use a relatively simple mapping to “translate” it.

There are two popular interpretations of the previous assumption. The first, which is shown in Figure 1, is the Bayesian interpretation. Here N

Figure 1 Bayesian interpretation of the variability of acoustic models

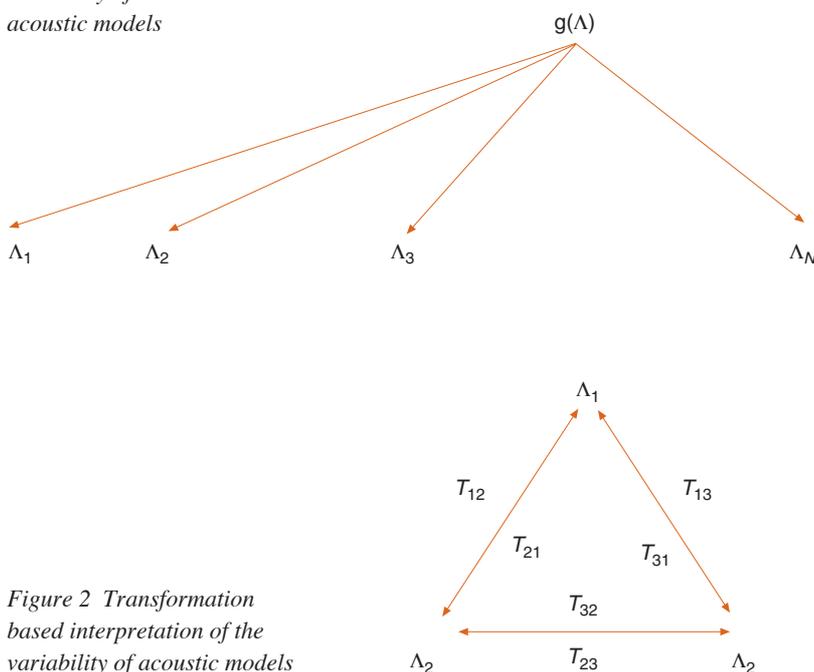


Figure 2 Transformation based interpretation of the variability of acoustic models

different HMMs represented by the parameter vectors, $\Lambda_1, \dots, \Lambda_N$, are assumed to be realizations of a *random variable* with probability density function $g(\Lambda)$. Those familiar with the Bayesian approach to statistics will immediately recognize $g(\Lambda)$ as the *prior distribution* of the HMM parameters. If we know this prior distribution, we already know something about “typical” HMMs and so less data should be needed to obtain a good estimate of a given HMM. This interpretation is the basis of the MAP adaptation approach which will be described later. Although MAP adaptation as it is presented in [3] is known to exhibit very slow convergence in terms of the amount of adaptation data needed, this criticism should not apply to MAP adaptation in general as the use of different correlation structures for $g(\Lambda)$ should allow for rapid adaptation.

Another interpretation is shown in Figure 2. According to this illustration any model Λ_i corresponding to a domain i , can be transformed into a new model Λ_j for a domain j using a transformation T_{ij} . If T_{ij} is assumed to be parametric we can estimate these parameters using our adaptation data. The exact form of these transformations are unknown in the same way that the prior $g(\Lambda)$ is unknown under the Bayesian paradigm, and to compensate for this lack of knowledge the transformations are chosen to be flexible while still analytically tractable. The well-known *maximum likelihood linear regression* (MLLR) adaptation algorithm [4], in which an *affine transformation* is used to transform the mean vectors of the mixture densities, is probably the best known example and will be presented in more detail in section 6. Richer sets of transformations in the form of neural networks have also been investigated – see e.g. [5].

In the next section we will briefly discuss offline vs. online adaptation. Then some previous work on model adaptation that sets the stage for our work will be presented.

4 Offline vs. Online Adaptation

Offline and online adaptation refers to whether the adaptation process is performed prior to system deployment or if it is done during recognition, respectively. Offline (or batch) adaptation is the simplest approach, since if we know that the HMM is to be used for e.g. a specific speaker or acoustic condition, then relevant data can be collected and the model adapted using any convenient algorithm. For most tasks where offline adaptation is feasible, the complexity of the algorithm used is secondary. An important goal is to achieve acceptable performance using as little adaptation data as possible, as collecting the data can be both expensive and tedious.

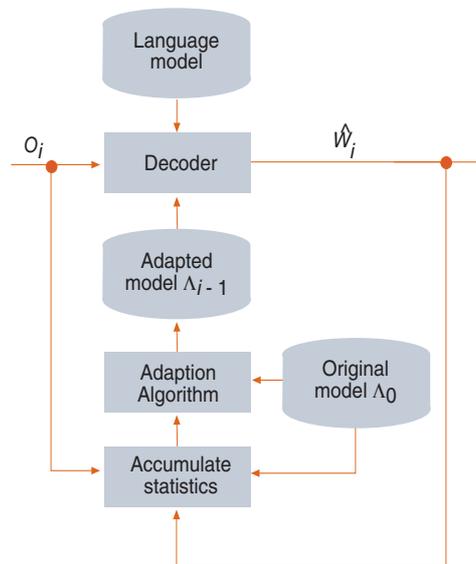


Figure 3 Online adaptation while keeping the initial model. The statistics of the adaptation speech are accumulated for every new utterance and used to adapt the initial model

Online adaptation, on the other hand, often imposes stricter conditions than its offline counterpart, as the amount of data available is more limited and the computational complexity can not be too high. Also, depending on the adaptation algorithm used, multiple recognition passes may be necessary, as well as keeping multiple models in memory. Two general approaches to online adaptation will be reviewed here.

The first approach is illustrated in Figure 3. For utterance number i , let O_i be the acoustic signal and \hat{W}_i the decoded string hypothesis found using the current model, Λ_{i-1} . Using this string hypothesis we obtain the observation statistics using the original HMM, Λ_0 . Depending on the form of the statistics, these can be accumulated for every new utterance. Finally, the adaptation algorithm makes use of these statistics to adapt the original model, yielding a new model, Λ_i . If the adaptation algorithm accumulates the statistics, this procedure should converge to the speaker/condition dependent model, provided that the new utterances come from the same speaker/condition. Of course, if the mismatch is so pronounced that the recognized string has no resemblance to the true utterance, the performance may very well be degenerating further. In such cases it is very important to constrain the flexibility of the adaptation algorithm.

The second approach is based on the recursive use of posterior distributions. Let

$\mathbf{O}_i = \{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_i\}$ be a sequence of independent utterances, and denote the corresponding sequence of recognition hypotheses by

$\hat{W}_i^i = \{\hat{W}_{1,\kappa}^i, \hat{W}_i^i\}$. We can then write the posterior probability of the model Λ as

$$\begin{aligned} & p(\Lambda | \mathbf{O}_i^i, \hat{W}_i^i) \\ &= \frac{p(\mathbf{O}_i | \mathbf{O}_i^{i-1}, \hat{W}_i^i, \Lambda) p(\mathbf{O}_i^{i-1}, \hat{W}_i^i, \Lambda)}{p(\mathbf{O}_i, \mathbf{O}_i^{i-1}, \hat{W}_i^i)} \\ &= \frac{p(\mathbf{O}_i | \mathbf{O}_i^{i-1}, \hat{W}_i^i, \Lambda) p(\Lambda | \mathbf{O}_i^{i-1}, \hat{W}_i^i) p(\mathbf{O}_i^{i-1}, \hat{W}_i^i)}{p(\mathbf{O}_i | \mathbf{O}_i^{i-1}, \hat{W}_i^i) p(\mathbf{O}_i^{i-1}, \hat{W}_i^i)} \\ &= \frac{p(\mathbf{O}_i | \hat{W}_i^i, \Lambda) p(\Lambda | \hat{W}_i^i, \mathbf{O}_i^{i-1})}{p(\mathbf{O}_i | \hat{W}_i^i)} \end{aligned} \quad (5)$$

where the last step is due to the independence assumption of the utterances. Note that any reference to the hyper-parameters has been dropped to simplify the equations. The adapted model can now be estimated in a MAP sense; that is, as the first mode of the posterior probability of the model,

$$\hat{\Lambda} = \arg \max_{\Lambda} p(\Lambda | \mathbf{O}_i^i, \hat{W}_i^i). \quad (6)$$

The interesting part of this estimate is that the prior used is the posterior probability of the model given the last $i - 1$ utterances,

$$p(\Lambda | \mathbf{O}_i^{i-1}, \hat{W}_i^{i-1}).$$

When the next utterance comes along we can adapt the model again, but now using our current posterior probability,

$$p(\Lambda | \mathbf{O}_i^i, \hat{W}_i^i),$$

as our prior. We see that the prior will be evolving as we process the utterances. If the speaker and acoustic condition remain unchanged, $p(\Lambda | \mathbf{O}_i^i, \hat{W}_i^i)$ will

become increasingly concentrated around the posterior mode, guaranteeing convergence. More details on MAP estimation can be found in the next section.

5 Bayesian Adaptation Techniques

We will in this section present some previous work on speaker adaptation and robust recognition where the key assumption has been the existence of a prior distribution for the HMM parameters.

5.1 MAP Adaptation

MAP adaptation is based on the Bayesian paradigm where the HMM parameters, Λ , are assumed to be distributed according to a prior distribution $g(\Lambda | \Phi)$. Here Φ is the hyper-parameter vector that together with the functional form of the prior distribution represents our knowledge about Λ . The complete knowledge is usually unobtainable in all but very special cases. Thus, most of the time the best one can do is to choose a functional form that is tractable, while still flexible enough to be able to capture the essential features of the “true” prior. In the following we will assume that our prior knowledge can be reflected using the proper choice of Φ . It should also be noted that in the strictest sense, MAP adaptation does not perform adaptation by changing an existing model. Instead, all the available information about the HMMs we want to find is represented in the form of a probability distribution, our prior, which allows us to pick the *a posteriori* most probable model with respect to the adaptation data.

Given an utterance \mathbf{O} and its transcription W , the adapted model can be obtained by finding the model that maximizes the *posterior probability*,

$$\begin{aligned} \Lambda_{\text{MAP}} &= \arg \max_{\Lambda} g(\Lambda | \mathbf{O}, W, \Phi) \\ &= \arg \max_{\Lambda} p(\mathbf{O} | W, \Lambda) g(\Lambda | \Phi) \end{aligned} \quad (7)$$

This approach was first investigated for adapting the mean and variance parameters in a small vocabulary recognizer [6]. A more formal treatment was given later in [3], in which the formulation was extended to include adaptation of the mixture weights and state transition probabilities. Later extensions include MAP adaptation for discrete HMMs (DHMM) and semi-continuous, or the tied-mixture, HMMs (SCHMM) [7].

The first problem to be solved is to determine the actual form of the prior $g(\Lambda | \Phi)$. As the HMM does not belong to the exponential family of distributions, it has no sufficient statistics and so there exists no posterior distribution of fixed dimension [8]. This again excludes the use of conjugate prior – posterior pairs [9]. However, it

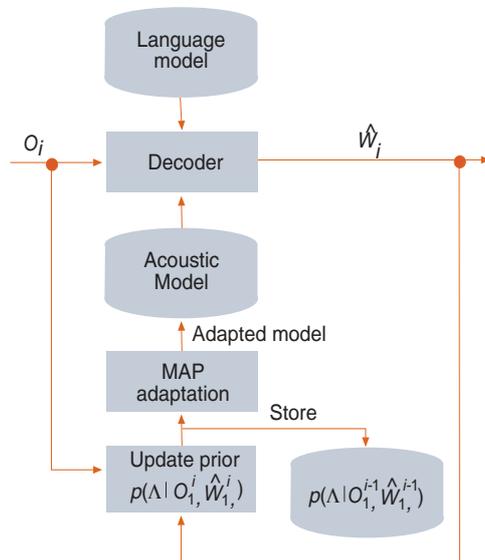


Figure 4 Online adaptation using prior evolution. New models are estimated in a MAP sense for every new utterance using the posterior probability density estimated for the previous utterance as the prior

is shown in [3] that conjugate priors exist for all the HMM parameters given the *complete data*. There, Dirichlet densities were used as priors for the state transition probabilities and the mixture weights, and normal-Wishart densities as priors for the mean vectors and covariance matrices of the multivariate normal distributions (see [10] for details on these distributions). Let $g(\Lambda | \Phi)$ denote the product of all these individual prior probability densities. According to [11] we can now estimate the posterior mode of $p(\Lambda | \mathbf{O}, \Phi)$ using the EM algorithm by augmenting the auxiliary function, $Q(\bar{\Lambda} | \Lambda)$, by $\log g(\Lambda | \Phi)$ during the maximization step.

Finally, the hyper-parameters have to be determined. In the statistics literature there are several formal approaches to the prior specification, but the prevailing one used in conjunction with MAP adaptation is based on the *empirical Bayes* paradigm, where the prior parameters are estimated using an independent training set. Using this approach the parameters of the prior distribution can be estimated directly from the training data. Typically the first and second modes are estimated as follows: The first mode of the prior is chosen so that

$$\Lambda_0 = \arg \max_{\Lambda} g(\Lambda | \Phi). \quad (8)$$

is some already known model, typically the model estimated in an ML sense using the training data. This choice is intuitively sound in the sense that when no adaptation data is available the MAP estimate still yields a useful model. The second mode is harder to estimate. One straightforward approach is to train several HMMs using different sets of training data and then estimate the second mode of the prior from this set of HMMs. However, this is hardly ever feasible, both due to the computations involved to create the number of models needed to achieve good estimates, as well as the problem of acquiring relevant data. In principle the second mode should reflect the variability of HMMs, but in practice the choice of parameters depend upon the task at hand. For example, using priors highly concentrated around the first mode implies that we “trust” our original model and to make a significant change more adaptation data is needed. This would be reasonable if we know a priori that the adaptation data is noisy.

5.2 Extended MAP Adaptation

The form of the prior chosen in [3] enables us to adapt our HMMs in a consistent and robust manner. Unfortunately, the independence assumption of the parameters of different sub-words implied by the simple correlation structure of the prior requires that all the different acoustic conditions,

for example every phoneme in every possible context, have to occur in the adaptation data for the whole of the model to be adapted. Ideally, the correlation between each model parameter should be known and available, but as an HMM used in a large vocabulary recognizer can contain several million parameters this is highly infeasible to store and use, not to mention the difficulties involved in the estimation of these parameters. However, the problem can be alleviated using some approximations. A class of algorithms often referred to as *extended MAP* (EMAP) [12], only considers the correlation between pairs of mean vectors under the assumption that they have a joint Gaussian distribution. Formally, if $\tilde{\mu} = [\mu_1, \dots, \mu_K]^T$ is a concatenation of all mixture means for a speaker dependent model and $\tilde{\mu}_0 = [\mu_1^0, \dots, \mu_K^0]^T$ is the concatenation of the mean vectors from the speaker independent model, the correlation matrix S_0 is given as

$$S_0 = E [(\tilde{\mu} - \tilde{\mu}_0)(\tilde{\mu} - \tilde{\mu}_0)^T] \quad (9)$$

The global correlation matrix can be considered a $K \times K$ block matrix of smaller correlation matrices $S_{O,J} = E [(\mu_i - \mu_i^0)(\mu_j - \mu_j^0)^T]$. In practice this matrix can be estimated using speaker dependent models, provided the model size is moderate. Using a prior probability density incorporating this correlation information we can update all the mixture mean vectors on the basis of their correlation with mean vectors corresponding to the adaptation data.

For large models the above approach turns out to be infeasible. One way to reduce the complexity is to share the correlations across clusters of Gaussians [13, 14]. In [15] even further simplifications were introduced by not considering a full correlation structure, but rather a single correlation scalar ρ_{ij} .

5.3 Online MAP With Prior Evolution

To complete our treatment of MAP adaptation, we now discuss some of the issues involved when using online adaptation with prior evolution. As mentioned previously in section 4, for every new utterance \mathbf{O}_i and its hypothesized

transcription \hat{W}_i , the MAP estimate of the new model is

$$\Lambda_i = \arg \max_{\Lambda} p(\mathbf{O}_i | \hat{W}_i, \Lambda) p(\Lambda | \mathbf{O}_1^{i-1}, \hat{W}_1^{i-1}),$$

that is, the model maximizing the posterior distribution. As the posterior distribution is sup-

posed to serve as our prior the next time we estimate a new model, we should look at its functional form in more detail. Using Bayes theorem the posterior distribution can now be written

$$\begin{aligned} & p(\Lambda | \mathbf{O}_i^j, \hat{W}_i^j) \\ &= \frac{p(\mathbf{O}_i^j | \hat{W}_i^j, \Lambda) p(\Lambda | \mathbf{O}_i^{j-1}, \hat{W}_i^{j-1})}{p(\mathbf{O}_i^j | \hat{W}_i^j)} \\ &= \frac{\sum_{Q \in \mathcal{Q}} \sum_{K \in \mathcal{K}} p(\mathbf{O}_i^j, Q, K | \hat{W}_i^j, \Lambda) p(\Lambda | \mathbf{O}_i^{j-1}, \hat{W}_i^{j-1})}{p(\mathbf{O}_i^j | \hat{W}_i^j)} \quad (10) \end{aligned}$$

where \mathcal{Q} and \mathcal{K} are the sets of all feasible state and mixture sequences, respectively. It is clear from (10) that the number of terms in the functional form increases at an exponential rate as new posterior densities are estimated. This problem has been addressed in [16] using the theory of Quasi-Bayesian sequential estimation [17].

Here the true posterior probability density $p(\Lambda | \mathbf{O}_i^j, \hat{W}_i^j)$ is approximated by a density

$g(\Lambda | \Phi_i)$ from a family \mathcal{G} of tractable densities. The most convenient approximation is to use the family of densities containing the prior, allowing the prior evolution to consist of a relatively simple parameter update.

6 Transformation Based Model Adaptation

The principle behind transformation based model adaptation is shown in Figure 2. The transformations we are interested in are all parametric, and so to adapt a given HMM using a specific transformation its parameters have to be estimated first. Usually this is done in a maximum likelihood sense in that we find the transformation of the model that maximizes the likelihood of our adaptation data.

More formally, a speaker independent model, Λ_{SI} , is transformed into a new model, $\Lambda_A = T_\eta(\Lambda_{SI})$, η being the transformation parameters, using the adaptation data $\{\mathbf{O}, W\}$. Doing this in a maximum likelihood sense means that

$$\eta_{ML} = \arg \max_{\eta} p(\mathbf{O} | T_\eta(\Lambda_{SI}), W). \quad (11)$$

If we have prior information of the speaker or the acoustic conditions we want to adapt our model to, we can formulate this knowledge in the form of a prior probability density $g(\eta | \phi)$, where ϕ are the hyper-parameters. This enables us to estimate the transformation parameters in a MAP sense, that is

$$\eta_{MAP} = \arg \max_{\eta} p(\eta | \mathbf{O}, \Lambda_{SI}, W) \quad (12)$$

$$= \arg \max_{\eta} p(\mathbf{O} | T_\eta(\Lambda_{SI}), W) g(\eta | \phi) \quad (13)$$

6.1 Bias, Affine and Nonlinear Transformations

One of the simplest transformations previously reported is the use of a bias term for the mean

vectors, $\hat{\boldsymbol{\mu}}_m = \boldsymbol{\mu} + \mathbf{b}_{r(m)}$, where $r(m)$ maps each

Gaussian mixture component into a cluster.

Sharing the bias across several mixtures enables us to adapt the whole cluster even if only one component is observed in the adaptation data.

In [18] this approach was investigated together with simple scaling transform of the mixture

covariance, $\hat{\Sigma}_m = \alpha_{r(m)} \Sigma_m$. Both the bias and

scaling parameters were estimated in the maximum likelihood sense using the EM algorithm.

A hybrid adaptation scheme based on a combination of the techniques described here and MAP adaptation was reported in [19], showing that the joint technique outperformed any of the individual approaches.

A slightly more complex approach was presented in [20]. Here an *affine transformation* (the affine transformations will be covered in more detail when we discuss MLLR adaptation in section 6.2) was used to adapt the mixture mean vectors using the relation

$\hat{\boldsymbol{\mu}}_m = \mathbf{A}_{r(m)} \boldsymbol{\mu}_m + \mathbf{b}_{r(m)}$. Here $\mathbf{A}_{r(m)}$ is a matrix and $\mathbf{b}_{r(m)}$ is a bias vector. The corresponding mixture covariance matrices are adapted as

$\hat{\Sigma}_m = \mathbf{A}_{r(m)} \Sigma_m \mathbf{A}_{r(m)}$ using the same linear

transformation as used with the mean vector. It should be mentioned that this approach is very close to the MLLR adaptation algorithm. This approach to speaker adaptation has also been used in conjunction with the MAP adaptation approach, as a hybrid approach [21].

More general transformations have also been investigated. In [22, 5] neural networks were used to implement a nonlinear mapping of mix-

ture mean vectors, $\hat{\boldsymbol{\mu}}_m = g_\eta(\boldsymbol{\mu}_m)$, where η repre-

sents the neural network. The neural network weights are estimated in the maximum likelihood sense using the EM algorithm, but unlike the case of the simple bias and affine transformation there is no closed form solution available for maximizing the likelihood in the M-step.

However, as long as the auxiliary function

$Q(\bar{\eta} | \eta)$ increases every iteration, $Q(\eta_{i+1} | \eta_i) >$

$Q(\eta_i | \eta_i)$, the likelihood of the adaptation data will increase monotonously. Thus, any suitable gradient search method can be applied.

Next we will discuss the MLLR approach in some detail for its popularity and effectiveness in model adaptation. The structural MAP (SMAP) approach will also be reviewed.

6.2 Maximum Likelihood Linear Regression

The idea behind maximum likelihood linear regression (MLLR) [23] is that model adaptation can be achieved by applying a parametric transformation to the model parameters. The transformation used in MLLR is an *affine transformation*. This is a map $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ of the form

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad (14)$$

where \mathbf{A} is a matrix and \mathbf{b} is a bias vector. In the rest of this work we will adhere to the literature and write this mapping as

$$\mathbf{W}\tilde{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad (15)$$

where $\mathbf{W} = [\mathbf{A} \ \mathbf{b}]$ and $\tilde{\mathbf{x}}$ is the vector \mathbf{x} augmented by a one. The affine mapping is applied to the mixture component mean vectors only, so that

$$\hat{\boldsymbol{\mu}}_m = \mathbf{W}\tilde{\boldsymbol{\mu}}_m. \quad (16)$$

The transformation \mathbf{W} is estimated in the maximum likelihood sense using the adaptation data. Let us for simplicity's sake assume that we are using a single, global transformation shared across all mixture component mean vectors. The likelihood of the adaptation data can then be written

$$p(\mathbf{O}|\mathbf{W}, \mathbf{A}, \mathbf{W}) = \sum_{Q \in \mathcal{Q}} \sum_{K \in \mathcal{K}} p(\mathbf{O}, Q, K | \mathbf{W}, \mathbf{A}, \mathbf{W}), \quad (17)$$

where \mathcal{Q} and \mathcal{K} are state and mixture occupation sequences, respectively. Formally we can write the maximum likelihood estimate of \mathbf{W} as

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} p(\mathbf{O} | \mathbf{W}, \mathbf{A}, \mathbf{W}). \quad (18)$$

Because this is a missing data problem \mathbf{W} cannot be found directly and so the EM-algorithm will be used. The auxiliary function

$$Q(\mathbf{W} | \mathbf{W}) = \sum_{Q \in \mathcal{Q}} \sum_{K \in \mathcal{K}} p(\mathbf{O}, Q, K | \mathbf{W}, \mathbf{A}, \mathbf{W}) \log p(\mathbf{O}, Q, K | \mathbf{W}, \mathbf{A}, \mathbf{W}) \quad (19)$$

is maximized over \mathbf{W} , yielding a set of $p(p+1)$ equations in $p(p+1)$ unknowns, p being the dimension of the mixture mean vectors. Fortunately, for HMMs containing diagonal covariance matrices we can reorganize these equations and solve p sets of $p+1$ equations in $p+1$ unknowns, where each set corresponds to one

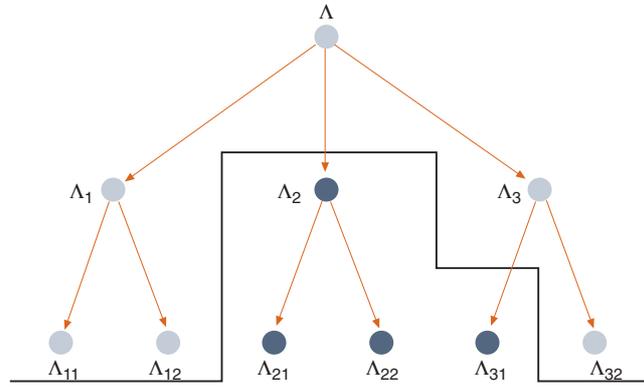


Figure 5 A tree structure of HMM parameters for use with multiple MLLR transformations. The line illustrates a cut in the tree where the transformations that have sufficient adaptation data are separated from the transformations that cannot be estimated robustly

row of the matrix \mathbf{W} . This alleviates the inevitable problem of ill-conditioned systems of linear equations somewhat, but not completely. The actual derivation of these linear equations is deferred to a later chapter, where the use of prior information will be incorporated.

A natural extension to the use of this global transformation is to apply separate transformations to different parts of the model. A very convenient approach is to organize the model parameters in a tree structure like the one shown in Figure 5. Here every node is the union of the disjoint sets of parameters in its child nodes. Using this structure one can also choose approximately the number of transformations according to the adaptation data available. The top node can naturally be estimated using all the adaptation data, while its child nodes will have to do with increasingly smaller fractions as we go down the tree. Setting a hard threshold enables us to consider only those transformations that can be estimated robustly. The tree itself can be constructed using either expert knowledge or an unsupervised clustering approach. An example of the former is to use phonetic questions to build a tree with binary splits.

A weakness of the MLLR approach is the sensitivity of the performance with respect to the number of transformations used. Using too many transformations will inevitably yield systems of linear equations that are very ill-conditioned, while using too few leads to a suboptimal performance. Although it is possible to tune the performance using thresholds on the minimum number of observations or setting a maximum number of transformations, this is a very crude way to control the smoothness of the global map that does the adaptation.

A natural extension to MLLR is to adapt the covariance matrices of the mixture densities. In [24] the adaptation is performed by setting the covariance matrix equal to

$$\hat{\Sigma}_m = \mathbf{B}_m^T \hat{\mathbf{H}}_m \mathbf{B}_m, \quad (20)$$

where \mathbf{B}_m is the inverse of the Cholesky factor of Σ_m^{-1} and $\hat{\mathbf{H}}_m$ is estimated in a maximum likelihood sense. This method has been reported to give a small, but significant, performance increase.

One extension of the basic MLLR formulation that can improve the robustness of the estimation process is reviewed here. In [25] the MLLR framework was extended to a MAP formulation using prior distributions on the transformation from the family of elliptically symmetric distributions [26], which yields a closed form solution to the maximization step of the EM algorithm. This approach led to an increase in performance compared to MLLR.

6.3 Structural Maximum a Posteriori Adaptation

Structural maximum a posteriori (SMAP) adaptation as published in [27, 28, 29] is not a Bayesian approach in the sense that we defined in Section 5. SMAP models the mismatch between the SI model and the test environment, and then uses the mismatch model to transform the SI model parameters. The mismatch is modeled as follows. Let

$$\mathbf{y}_{mt} = \Sigma_m^{-1/2}(\mathbf{x}_t - \mu_m)$$

for every mixture component m and adaptation feature vector \mathbf{x}_t . The idea is now that $\mathbf{y} \sim \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{I})$ if there is no mismatch. If this is not true, however, we would expect $\mathbf{y} \sim \mathcal{N}(\mathbf{y}; \nu, \eta)$. This distribution is referred to as a *normalized pdf*.

Given the transformed observations, \mathbf{y}_{mt} , the maximum likelihood estimates of the model parameters are

$$\hat{\nu} = \frac{\sum_{t=1}^T \sum_{m=1}^M \gamma_t(m) \mathbf{y}_{mt}}{\sum_{t=1}^T \sum_{m=1}^M \gamma_t(m)}$$

$$\hat{\eta} = \frac{\sum_{t=1}^T \sum_{m=1}^M \gamma_t(m) (\mathbf{y}_{mt} - \hat{\nu})(\mathbf{y}_{mt} - \hat{\nu})^T}{\sum_{t=1}^T \sum_{m=1}^M \gamma_t(m)}$$

where $\gamma_t(m)$ is the probability of an observation vector coming from mixture m at time t . Assuming that the estimates $\{\hat{\nu}, \hat{\eta}\}$ are known, the adaptation step consists of transforming the mixture component mean and variance parameters. This is achieved using the relations,

$$\hat{\mu}_m = \mu_m + \Sigma_m^{1/2} \hat{\nu}$$

$$\hat{\Sigma}_m = \Sigma_m^{1/2} \hat{\eta} \Sigma_m^{1/2}.$$

In [29] the model parameters were organized in a tree structure of mixture component clusters, similar to the approach described in 6.2 for MLLR. The reasons for this were twofold. For one, this is a refinement of the mismatch model which in turn leads to more accurate modeling of the actual acoustic mismatch. Second, the tree structure can be used in conjunction with the hierarchical Bayes paradigm to yield MAP estimates.

7 Adaptive Training

The final “main direction” in adaptation that we will discuss is adaptive training. In principle adaptive training encompasses all techniques that use information embedded in the training data to facilitate a more efficient adaptation. A typical assumption used in adaptive training is that the variability that we will encounter when deploying the recognition system will be present, to some extent, in the training data. A trivial example is gender dependent models, in which one model is trained separately for each gender and the model selection is done during the recognition phase. This approach has been shown to exhibit a significant performance improvement compared to gender independent models [30].

Another approach which in some sense incorporates the gender variations while allowing for a finer modeling, is *vocal tract length normalization* (VTLN) [31]. The primary effect of having different vocal tract lengths comes down to a warping of the frequency. That is, to match the spectrum of the “same” sound coming from two different speakers, the frequency axis of one of the speakers will have to be warped by a factor α . The idea behind VTLN is that if this variation, which can be considered irrelevant as far as speech recognition is concerned, can be removed, we will be able to model the normalized speech better than if the variation has to be modeled as well. When speech from a new speaker is to be recognized, the warping factor is determined in a maximum likelihood sense and the speech normalized during the feature extraction phase. It has also been shown that VTLN and other adaptation approaches like MLLR are complementary, although the performance increase is sub-additive [32].

The two previous examples, while trivial, illustrate two separate directions in adaptive training. When using gender dependent models we try to identify speakers having some dissimilarities that are best modeled separately, while VTLN tries to normalize the speakers by removing the variabilities which are irrelevant to the information crucial to speech recognition. Next we will present three approaches of somewhat higher complexity that are showing some promise.

These are speaker adaptive training (SAT), eigenvoices and cluster adaptive training (CAT).

7.1 Speaker Adaptive Training

In speaker adaptive training (SAT) the normalization principle of VTLN is taken one step further [33, 34]. Its major assumption is the existence of a so-called *compact model*, Λ_C , that contains all relevant acoustic information for recognition, and that every speaker r is related to this model through a transformation, T_r . Typically T_r is in the form of one or more affine transformations as in MLLR. If $\mathcal{T} = \{T_1, \dots, T_R\}$ is the set of all transformations with respect to all speakers, the maximum likelihood estimate of the compact model is

$$\{\Lambda_C, \mathcal{T}\} = \arg \max_{\Lambda, \mathcal{T}} \prod_{r=1}^R p(\mathbf{O}^{(r)} | T_r, \Lambda_C, W^{(r)}), \quad (21)$$

where $\{\mathbf{O}^{(r)}, W^{(r)}\}$ is the training data from speaker r . The compact model and the transformations can be found using the EM algorithm. The resulting compact model is reported to give a better performance when adapted to a test speaker than to a speaker independent model adapted in a similar way. Also, a small performance increase compared to the speaker independent model is reported when no adaptation is used at all.

7.2 Cluster Adaptive Training

The main assumption behind cluster adaptive training is that there exist clusters of speakers exhibiting similar acoustic properties. Any unseen speaker is assumed to belong to one of these clusters and so we should expect a performance increase if we could successfully identify this cluster and perform the recognition using a “cluster-dependent” model.

In [35] a *canonical model* is introduced and used to model different clusters of speakers as one. This is made possible by introducing a matrix of

C canonical means, $\mathbf{M}_m = [\boldsymbol{\mu}_m^{(1)}, \dots, \boldsymbol{\mu}_m^{(C)}]$, for

every mixture component m . Given a set \mathcal{M} such matrices for all mixtures m , we can estimate the speaker dependent means, $\{\boldsymbol{\mu}_m\}$, for a speaker r using

$$\boldsymbol{\mu}_m = \mathbf{M}_m \boldsymbol{\lambda}_r, \quad (22)$$

where $\boldsymbol{\lambda}_r$ is the *cluster weight vector* for speaker r . An alternative formulation is

$$\boldsymbol{\mu}_m = \mathbf{M}_m \boldsymbol{\lambda}_r + \mathbf{b}, \quad (23)$$

where \mathbf{b} is a bias term and is typically equal to the speaker independent mean.

Whichever formulation is used, we have to train the canonical model by estimating the \mathbf{M}_m matrix for every mixture m . Then, using this canonical model we can estimate a speaker dependent model for any speaker r by finding the cluster weight vector $\boldsymbol{\lambda}_r$ in a maximum likelihood sense. As $\boldsymbol{\lambda}_r$ has the same dimensionality as the number of clusters C , we expect to be able to estimate this vector using very little adaptation data. The performance ultimately depends on how “close” the speaker is to the clusters, or any linear combination thereof.

7.3 Eigenvoices

The term eigenvoices ([36]) was inspired by the use of eigenfaces in the face recognition research community [37]. However, the connection between eigenvoices and real human voices is still somewhat vague as the eigenvoices are derived from the parameter vectors of HMMs.

To find the eigenvoices a speaker dependent model Λ_r will have to be trained for every speaker r . Then, all the mean vectors of each model Λ_r are concatenated into a single *super-vector* \mathbf{m}_r , resulting in a set of R very long vectors. On the basis of this collection of vectors *principal component analysis* (PCA [38]) is performed, and the $K \ll R$ eigenvectors corresponding to the K largest eigenvalues are stored. A speaker independent model is also trained and a super-vector consisting of the concatenation of the mixture means is stored.

For every new speaker a new model can now be found by estimating the mean vectors as a linear combination of the SI super-vector and the eigenvoice-vectors. This is constructed in a maximum likelihood sense, and so in this respect the use of eigenvoices has a lot in common with CAT. Eigenvoices have been shown to significantly improve the performance for some simpler tasks like the isolated letter recognition, when as little as a few seconds of adaptation material were available.

8 Summary

In this article we have reviewed the use of model adaptation as a way to alleviate mismatches between the training and the test conditions. We have also presented the principles behind model adaptation, as well as a summary of some important research directions.

The two major directions in model adaptation is the Bayesian approach employed in MAP and EMAP, and the transformation based approach used in for example MLLR. The two approaches have different strengths and weaknesses: Although there is no theoretical basis for this, in practice MAP adaptation needs large amounts of data to update every model parameter due to the

simple prior used. MLLR, on the other hand, can improve the model significantly using only a single utterance. However, when enough adaptation data is available MAP has the desirable property of convergence to the speaker dependent model. Also, the use of prior distributions allows for a smoother control of the learning rate. A third direction, which we have referred to as adaptive training, is also gaining in popularity. These approaches use the training data to model the variations that will occur on the test data.

Finally, it is clear that current model adaptation techniques are able to provide substantial performance increases given a moderate amount of adaptation data. But, if model adaptation is to be a practical technology, complexity and memory use need to be addressed. Although desktop computers are more than powerful enough to handle a single user at the time, more and more interest is directed at ASR applications for PDAs and cellular phones. There is also a need for online adaptation approaches that are robust with respect to adverse conditions. Model adaptation is still a rich field with respect to future research.

References

- 1 Siohan, O, Myrvoll, T A, Lee, C-H. Structural maximum a posteriori linear regression for fast HMM adaptation. *Computer Speech and Language*, 16 (1), 5–25, 2002.
- 2 Rosenfeld, R. Two decades of statistical language modeling : Where do we go from here. *Proceedings of the IEEE*, 88, Aug. 2000.
- 3 Gauvain, J-L, Lee, C-H. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2 (2), 1994.
- 4 Legetter, C J, Woodland, P C. Maximum likelihood linear regression for speaker adaptation of the parameters of continuous density hidden Markov models. *Computer Speech and Language*, 9 (2), 171–185, 1995.
- 5 Surendran, A C, Lee, C-H, Rahim, M. Non-linear compensation for stochastic matching. *IEEE Transactions on Speech and Audio Processing*, 7 (6), 643–655, 1999.
- 6 Lee, C H, Lin, C H, Juang, B H. A study on speaker adaptation of the parameters of continuous density hidden Markov models. *IEEE Transactions on Signal Processing*, 39 (4), 806–814, 1991.
- 7 Huo, Q, Chan, C, Lee, C-H. Bayesian adaptive learning for the parameters of hidden Markov model for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 3, 334–345, 1995.
- 8 Koopman, B O. On distributions admitting a sufficient statistic. *Trans. Amer. Math. Soc.*, 39, 399–409, 1934.
- 9 Robert, C P. *The Bayesian Choice : a decision-theoretic motivation*. NY, Springer, 1994. (Springer texts in statistics) ISBN 0-387-94296-3, 3-540-94296-3
- 10 DeGroot, M H. *Optimal Statistical Decisions*. NY, McGraw-Hill, 1970. ISBN 0-07-016242-5
- 11 Dempster, A J, Laird, N M, Rubin, D B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (1), 1–38, 1977.
- 12 Lasry, M J, Stern, R M. A posteriori estimation of correlated jointly Gaussian mean vectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 (4), 530–535, 1984.
- 13 Zavaliagos, G et al. Adaptation algorithms for large scale HMM recognizers. In: *EuroSpeech '95*, Madrid, Spain, Sep. 1995.
- 14 Chen, S S, DeSouza, P. Speaker adaptation by correlation. In: *EuroSpeech '97*, Rhodes, Greece, Sep. 1997.
- 15 Huo, Q, Lee, C-H. On-line adaptive learning of the correlated continuous density hidden Markov model for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6 (4), 386–397, 1998.
- 16 Huo, Q, Lee, C-H. On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate. *IEEE Transactions on Speech and Audio Processing*, 5 (2), 161–172, 1997.
- 17 Smith, A F M, Makov, U E. A quasi-Bayes sequential procedure for mixtures. *Journal of the Royal Statistical Society*, 40 (1), 106–112, 1978.
- 18 Sankar, A, Lee, C-H. A maximum likelihood approach to stochastic matching for robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4 (3), 190–202, 1996.

- 19 Chien, J-T, Lee, C-H, Wang, H-C. A hybrid algorithm for speaker adaptation using map transformation and adaptation. *IEEE Signal Processing Letters*, 4 (6), 167–170, 1997.
- 20 Digalakis, V V, Rtsichev, D, Neumeyer, L G. Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, 3 (5), 357–366, 1995.
- 21 Digalakis, V V, Neumeyer, L G. Speaker adaptation using combined transformation and Bayesian methods. *IEEE Transactions on Speech and Audio Processing*, 4 (4), 294–300, 1996.
- 22 Abrash, V et al. Acoustic adaptation using nonlinear transformations of HMM parameters. In: *Proc. IEEE ICASSP-96* [39].
- 23 Leggetter, C J, Woodland, P C. Speaker adaptation of continuous density HMMs using multivariate linear regression. In: *Proc. ICSLP*, Yokahama, Japan, Sep. 1994.
- 24 Gales, M J F, Woodland, P C. *Variance compensation within the MLLR framework*. Cambridge University Engineering Department, Feb. 1996. (Tech. rep.)
- 25 Siohan, O, Chesta, C, Lee, C-H. Hidden Markov model adaptation using maximum a posteriori linear regression. In: *Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, 1999.
- 26 Gupta, A K, Varga, T. *Elliptically Contoured Models in Statistics*. Dordrecht, Kluwer, 1993. (Mathematics and its applications) ISBN 0-7923-2115-4
- 27 Shinoda, K, Lee, C-H. Structural map speaker adaptation using hierarchical priors. In: *Proceedings of IEEE Workshop on Speech Recognition and Understanding*, Santa Barbara, CA, USA, 1997.
- 28 Shinoda, K, Lee, C-H. Unsupervised adaptation using structural Bayes approach. In: *Proc. IEEE ICASSP-98*, Seattle, WA, May 1998.
- 29 Shinoda, K, Lee, C-H. A structural Bayes approach to speaker adaptation. *IEEE Transactions on Speech and Audio Processing*, 9 (3), 276–287, 2001.
- 30 Woodland, P C et al. Large vocabulary continuous speech recognition using HTK. In: *Proc. IEEE ICASSP-94*, Adelaide, Australia, Apr. 1994.
- 31 Lee, L, Rose, R. A frequency warping approach to speaker normalization. *IEEE Transactions on Speech and Audio Processing*, 6 (1), 49–60, 1998.
- 32 Pye, D, Woodland, P C. Experiments in speaker normalization and adaptation for large vocabulary speech recognition. In: *Proc. IEEE ICASSP-96* [39].
- 33 Anastasakos, T et al. A compact model for speaker-adaptive training. In: *Proc. ICSLP*, Philadelphia, PA, Oct. 1996.
- 34 Anastasakos, T, McDonough, J, Makhoul, J. Speaker adaptive training : A maximum likelihood approach to speaker normalization. In: *Proc. IEEE ICASSP-97*, Munich, Germany, Apr. 1997.
- 35 Gales, M J F. Cluster adaptive training of hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 8 (4), 417–428, 2000.
- 36 Kuhn, R et al. Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8 (6), 695–707, 2000.
- 37 Kirby, M, Sirovich, L. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 (1), 103–108, 1990.
- 38 Anderson, T W. *An Introduction to Multivariate Statistical Analysis*, second ed. NY, John Wiley, 1984. ISBN 0-471-88987-3
- 39 *Proc. IEEE ICASSP-96*, Atlanta, Georgia, May 1996.

Pronunciation Variation Modeling in Automatic Speech Recognition

INGUNN AMDAL AND ERIC FOSLER-LUSSIER



Ingunn Amdal (38) graduated from the Norwegian University of Science and Technology (NTNU) with a *Siv.Ing.* (MSc) in 1989 and a *Dr.Ing.* (PhD) in 2002. In 1990 she started as research scientist at Telenor R&D working on loudspeaking telephones and acoustic echo-cancellation. She joined the newly formed speech technology group at Telenor R&D in 1994 working with automatic speech recognition, dialogue design, and database collection. The topic of her PhD thesis was pronunciation modelling and her current interests include design, representation and evaluation of spoken and multimodal dialogue systems, user tests and spontaneous speech processing.

ingunn.amdal@telenor.com



Eric Fosler-Lussier is currently a Visiting Research Scientist in the Electrical Engineering department of Columbia University, working on novel approaches to automatic speech recognition (ASR) and topic segmentation in recorded multi-party conversations. He received his PhD in 1999 from U. California, Berkeley; his area of dissertation research was dynamic pronunciation modeling for ASR. Subsequently, Fosler-Lussier was a Member of Technical Staff at Bell Labs, Lucent Technologies 2000–2002, conducting research on spoken dialogue system design, natural language call routing, pronunciation modeling, and language modeling for ASR.

fosler@ieee.org

Robust speech recognition is a critical research topic – systems must be able to handle a wide variation in types of speech to make speech technology more user-friendly. One major source of variation in speech is different speaking styles; handling this variation in user input is difficult for current state-of-the-art recognizers. Modeling pronunciation variation within the system can ameliorate the difficulties to some degree. Pronunciation variation can be modeled in different parts of the recognizer; in this presentation we focus on lexical adaptation (other articles in this issue of *Teletronikk* cover other types of robust modeling).

An overview of the methods used in pronunciation variation modeling by lexical adaptation will be given. First, the automatic speech recognition system will be explained briefly with a focus on the pronunciation lexicon. Then, the main distinction between pronunciation modeling methods, knowledge based or data-driven, is explained and illustrated with examples from selected work done in the field. Another distinction often made is modeling of the pronunciation variants directly or indirectly through pronunciation rules that make it possible to generalize knowledge or observations in a training set to unseen data. Finally, a section on confusability reduction is included.

1 Introduction

Early automatic speech recognition (ASR) systems only considered restricted speaking styles, i.e. careful articulation of isolated or connected words. The increased modeling capacities of current ASR systems also manage the looser articulation of continuous speech. Making speech technology based applications more widespread has several consequences for the demands on ASR systems:

New and larger vocabularies are needed when ASR systems are used in new domains. When the vocabulary is increased it is no longer feasible to select the pronunciation variants by hand. Pronunciations from various sources will often be combined and a decision must be taken whether to include one or more variants per word. The less restricted grammar of a large vocabulary speech recognition system will give more confusability, and more care must be taken in the selection of pronunciation.

When dialogues between humans and computers are more natural, the ASR must handle more conversational speech. Conversational speech is harder for ASR systems to recognize correctly, because of increased coarticulation and pronunciation variability, as well as less predictable language usage. Weintraub et al. [1] showed that a spontaneous speaking style is harder to recognize; when the same exact word sequences were recorded in a truly spontaneous, acted spontaneous, and read style, the ASR system performed much worse on spontaneous speech compared with the other two styles.

Large-scale deployment of systems may increase the amount of dialectal and accented speech that

a speech technology application may encounter.

A more international community increases the portion of non-native speakers in the general public; the substantial differences between native and non-native speech will challenge a natively trained ASR system. Even subtle differences that are easily handled by humans (e.g. Australian versus US English) can still cause problems for ASR.

The expertise of the user will change the speaking style used. Novice users will often hyperarticulate, especially when the system misrecognizes them (“Not Austin, I said BOSTON”). Expert users may become more relaxed, and engage with the system using a less formal speaking style. The type of variation seen will depend on the conversational style of prompts in a spoken dialogue system, as well as on the type of task: systems where many people call once or twice will encounter more novice speech, whereas personal dictation systems may encounter speech from experienced users.

Ideally, speech recognizers should handle these diverse speaking styles, (e.g. spontaneous speech, hyperarticulated speech, accents, dialects, and speech from users with different mother tongues). This kind of variation in user input is difficult to model and this is not solved for in current state-of-the-art recognizers.

Pronunciation modeling is by no means a new issue in the ASR community, early efforts are reported in e.g. [2] and [3]. Pronunciation variation modeling is still an important issue in ASR research, and overviews are for example given in [4] and [5]. More recently multilingual ASR has become an interest [6], which introduces new challenges for pronunciation modeling.

2 The Automatic Speech Recognition System

The recognition system may be divided into three main elements as shown in the three lines of Figure 1:

- 1 From speech via acoustic features to sub-word units: acoustic models
- 2 From sub-word units to words: lexicon
- 3 From words to sentences: language model

The three modules shown in Figure 1 constitute an automatic speech recognizer. For a recognizer operating in a real speech-based application, the user input will replace the “test speech” in the figure.

Mathematically the system can be described as a classifier. We observe a sequence (i.e. a feature vector representing speech) $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T)$ and want to find the word (sequence) W that maximizes the *a posteriori probability* (MAP) which will give us the *Bayes classifier*:

$$\hat{W} = \arg \max_W P(W|\mathbf{O}) = \arg \max_W \frac{p(\mathbf{O}|W)P(W)}{p(\mathbf{O})} \quad (1)$$

Since the observation \mathbf{O} is fixed and $p(\mathbf{O})$ is independent of W equation (1) reduces to:

$$\hat{W} = \arg \max_W p(\mathbf{O}|W)P(W) \quad (2)$$

$P(W)$ is generally called the probability of the *language model*, while $p(\mathbf{O}|W)$ is the probability density of the *acoustic model*.

Recent work [7, 8] has shown that the recognition process can be modeled with a sequence of finite-state transducers (FSTs). An abstract representation of the Viterbi decoding process might be given as:

$$\hat{W} = \text{bestpath}(A \circ T \circ P \circ L) \quad (3)$$

where \hat{W} is the sequence of words corresponding to the best recognition hypothesis, A is a finite state automaton (FSA) containing the set of acoustic scores computed from an input utterance, T is a context-dependent FST, containing a mapping from acoustic states to triphones, P is the pronunciation model FST, containing a mapping from triphones to words, L is the language model FSA, which contains N -gram statistics, and \circ is the composition operator. All of these finite state machines are typically weighted, with the costs derived from the probabilities of the particular linguistic model.

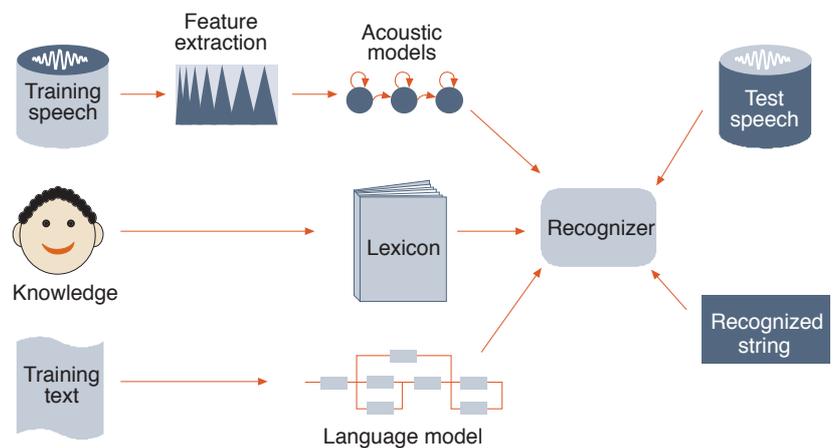


Figure 1 Automatic speech recognition system

3 How to Model Pronunciation Variation in ASR

Pronunciation variation modeling can be implemented in different parts of the speech recognizer shown in Figure 1. Different realizations of phonemes (allophonic variation) can for example be handled using either more acoustic units, more complex models (but fewer units), or more pronunciation alternatives. The allophonic variation can also be handled by adapting the models to one speaker or a more homogeneous subset of speakers.

The statistically based acoustic models of current ASR systems are capable of handling much of the variation seen in speech, including some pronunciation variation [9]. More complex acoustic models will, for example, handle many allophonic variations in a suitable way. Adaptation of the acoustic models is a successful method to make speaker-dependent recognizers with improved performance compared with speaker-independent recognizers. Task adaptation or adaptation to a group of speakers (e.g. dialect adaptation) is possible, but in general, the more homogeneous the adaptation target, the better the performance. Modeling large variations within the same model by broadening the distributions or adding more components in the Gaussian mixture will give more diffuse, overlapping models that may lead to increased confusability [10]. The success of this type of speaker adaptation also depends on the match between the actual pronunciations and the transcription used in adaptation.

Some of the pronunciation variation is caused by speaking style (dialects, non-native mother tongue, etc.), and may be better handled by careful design of the pronunciation dictionary, i.e. pronunciation modeling [9]. The most common way of dealing with pronunciation variation is to put several pronunciation alternatives in the ASR lexicon. These pronunciations are also often used to retranscribe the speech corpus

| Word | Pronunciation |
|-------------|-------------------------------|
| READ | r eh1 d |
| READ (2) | r iy1 d |
| READABILITY | r iy2 d ah0 b ih1 l ih0 t iy0 |
| READABLE | r iy1 d ah0 b ah0 l |
| READER | r iy1 d er0 |
| READER'S | r iy1 d er0 z |
| REALIZE | r iy1 l ay2 z |
| REALIZE (2) | r iy1 ah0 l ay2 z |
| RIGHT | r ay1 t |
| TOO | t uw1 |
| TWO | t uw1 |

Table 1 Examples of ASR lexicon entries

before a retraining of the acoustic models. Using the lexicon to capture speaker variation makes it possible to model several speakers simultaneously, thus using the same lexicon and the same acoustic models for all speakers.

High quality recognizers always include a language model, which is sometimes incorporated in pronunciation modeling techniques. For large vocabulary speech recognition, a well-designed language model may decrease the negative impact of a mismatch between the speaker and the acoustic models and explicit pronunciation modeling may be less important. If the speaking style we try to model has special language model characteristics, e.g. the hesitations and restarts of spontaneous speech, they may be incorporated directly into the language model [11].

One of the main challenges in pronunciation modeling is to know which variation we are attempting to model. The effects of the acoustic models, the lexicon, and the language model will interact, even the choices at the speech pre-processing stage will influence the variation modeling. Superfluous complexity or, even worse, adding contradicting changes, may result from modeling the same variation in several recognizer components. The two main techniques for capturing variation, acoustic model adaptation and lexicon adaptation, should be combined using the method that gives the best result: acoustic model adaptation for the pronunciation variation that can be described as allophonic, and lexicon adaptation for the more phonological variation like deletions and insertions. In this article, we focus on lexicon adaptation.

4 The ASR Lexicon

If units other than words are used for the acoustic models, we need a correspondence between the acoustic model units and the words in the vocabulary. When phones form the basic acoustic model units, this corresponds to, but is not necessarily equal to, a pronunciation dictionary. In the ASR community, the term *lexicon* is often used instead of dictionary for this link. The middle line in Figure 1 shows that the lexicon is often based on knowledge contrasting the optimization based on speech data and objective criteria that are used in the other parts of the recognizer. Up till now most ASR systems use only one canonical or a few pronunciations per word, and these pronunciations have typically been transcribed manually.

Some examples of entries in the CMU lexicon [12] are given in Table 1.¹⁾ A number after the word indicates a non-canonical pronunciation. The numbers after the vowels indicate stress; not all ASR lexica provide this information. Syllable information is also given in some ASR lexica. First of all we note that the word class is usually not indicated. A consequence of this is that there is no distinction between *homographs* (words with the same spelling but different meanings) with different pronunciations, such as the two tenses of “read” in Table 1. Often an ASR lexicon will have only one entry in such cases. For ASR lexica with multiple entries, several pronunciations can be given for one word, e.g. “realize”. There is usually only one entry for words with different senses, but identical spellings and pronunciations (*homonyms*). For example, “right” can mean both a direction and a notion of correctness (in addition to other senses²⁾). Homonymy can be a problem because the word sense information that could be useful for language modeling and semantic parsing is hidden. *Homophones* are words that have different meanings (and usually different spellings), but the same pronunciations. These words will have separate entries, as shown for “two” and “too”, but the ASR system must rely on the language model to resolve which word is recognized. This is the same as for human speech recognition, except that we usually have more contextual information available, such as the setting and theme for the spoken utterance and the identity of the speaker (this is a topic of pragmatics). In ASR, such knowledge is incorporated by using task and dialogue state dependent language models.

¹⁾ We use the ARPABET [13] for phonetic transcriptions.

²⁾ The Concise Oxford Dictionary lists 6 adjectives, 3 adverbs, 4 nouns, and 2 verbs for the word “right”.

When multiple pronunciations are used for each word, pronunciation probabilities may be used to inhibit confusions due to rare pronunciations. Pronunciation probabilities are often defined as part of the language model instead of the lexicon. From a hand-labeled part of the speech database Switchboard³⁾ 36 different pronunciations for “the” were found in the test set, and 38 different pronunciations in the training set. Only half of the variants found in the training set were also observed in the test set. The confusability caused by adding all observed variants can also be illustrated by the 35 different words that had the pronunciation [ax] (schwa) [15].

When adding more pronunciations for each word, we can make the word probability dependent on the pronunciation probability. The ASR classifier equation (2) can then be decomposed to include the pronunciations \mathcal{B} for the word W (often referred to as the base forms of W):

$$p(\mathbf{O}|W)P(W) = \sum_{B \in \mathcal{B}} p(\mathbf{O}|B, W)P(B|W)P(W) \approx \max_{B \in \mathcal{B}} [p(\mathbf{O}|B, W)P(B|W)P(W)] \quad (4)$$

The last line is the Viterbi approximation of using only the best pronunciation. The pronunciation probability $P(B|W)$ can be defined as a part of the language model and we get the language model probability $P(B|W)P(W)$.

It was shown in [16] that augmenting the recognizer lexicon with pronunciation variants found in a general-purpose lexicon gave small performance gains, and most for read native speech. Error analysis showed that the system using a single canonical pronunciation generated different errors than the one using pronunciation variants, although the word error rate was similar. For non-native speech, no improvement was observed for context-dependent acoustic models compared with context-independent models. This speaking style had the largest gain using speaker dependent acoustic model adaptation, but the performance was still far from the results for native speech. For spontaneous speech less improvement was observed by speaker adaptation than for read speech. Even if speaker adaptation was shown to give large improvements, the resulting performance was worse than for native read speech. To achieve results more comparable to native read speech for these two tasks, a combination of lexical and acoustic adaptation may be beneficial.

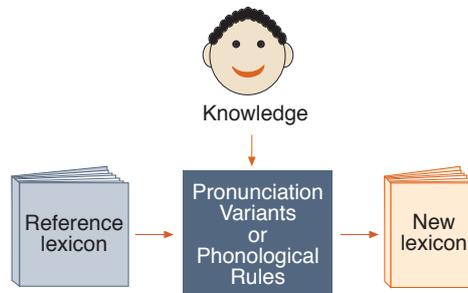


Figure 2 Knowledge-based lexicon adaptation

5 Lexicon Adaptation

There are two main directions in finding pronunciation variations, each involving different problems:

- 1 *Knowledge based methods*, where we try to find the best pronunciation rules by applying phonetic and linguistic knowledge as shown in Figure 2. The main problem occurs if the knowledge does not cover the variation we want to model. We may then have too many or too few variations and we may not know how frequent they are.
- 2 *Data-driven methods*, where we use databases of real speech to find the variations present as shown in Figure 3. The problem is that the variations based on a given database may give a result too specific for that database. One of the advantages is that we may compute probabilities for the variants, as opposed to the knowledge-based methods.

For both these methods we can distinguish between *direct* and *indirect* modeling. The pronunciation variants can either be derived directly for each word or indirectly by deriving pronunciation rules and using these rules to generate new pronunciations. Data-driven direct modeling limits us to model only words observed sufficiently many times in the adaptation set, whereas for indirect modeling (both for data-driven and knowledge based) care must be taken

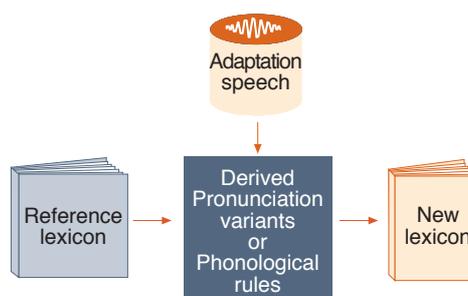


Figure 3 Data-driven lexicon adaptation

³⁾ Switchboard is a standard corpus of human-human telephone conversations in US English available from the Linguistic Data Consortium; a portion of it was phonetically transcribed by Greenberg et al. [14].

| Name | Rule |
|-------------|---|
| Reductions: | |
| Mid vowels | -stress [aa ae ah ao eh er ey ow uh] → ax |
| High vowels | -stress [iy ih uw] → ix |
| R-vowel | -stress [er] → axr |
| Syllabic n | [ax ix] n → en |
| Syllabic m | [ax ix] m → em |
| Syllabic l | [ax ix] l → el |
| Syllabic r | [ax ix] r → axr |
| Flapping | [tcl dcl][t d] → dx / V _ [ax ix axr] |
| Flapping-r | [tcl dcl][t d] → dx / V r _ [ax ix axr] |
| H-voicing | hh → hv / [+voice] _ [+voice] |

Table 2 Knowledge based phonological rules for US English, after [22]

in the generalization from the observed variation. If a certain variation appears in very different contexts in the adaptation data compared with the test data the generalization may not be valid. Variation observed in function words in the adaptation data may for example not be a variation appropriate for content words in the test data even if the phone context is the same. This can be illustrated by the function word “for” with the canonical pronunciation [f ao r] and the alternative pronunciation [f er]. This transformation is not equally probable for the noun “forest” with the canonical pronunciation [f ao r ah s t].

In the FST formalism given in equation (3), indirect modeling using phonological rules can be represented as an additional phone confusion FST C between the triphones and the lexicon [17]:

$$\hat{W} = \text{bestpath}(A \circ T \circ C \circ P \circ L) \quad (5)$$

The main reasons to use indirect modeling in a data-driven approach are:

- The vocabulary of the data used for rule derivation can be different from that of the test data. Rules help us generalize the variation seen in the adaptation data to words not present (“unseen words”).
- Rules depend on smaller segments than words and will occur more often, giving more reliable estimates.
- A possible extension to crossword rules will be easier.

Some of the pronunciation variation will be present across word boundaries [18]. Using rules makes the extension to cross-word pronunciation modeling easier, although multi-words make it

possible to model cross-word effects also when dealing directly with pronunciation variants [19]. *Multi-words* are new lexical items formed of several words, e.g. “going to” can be treated as one word to account for the pronunciation variant [g aa n ax].

In general, there has been a migration from knowledge-based methods to data-driven methods. In [20] it was shown that general-purpose lexica do not model spontaneous speech sufficiently well. Only 33 % of the pronunciations found in the hand-labeled part of Switchboard were present in the Pronlex dictionary [21]. The non-canonical pronunciations showed an 11 % increase in word error rate over canonically pronounced words. Another observation was that frequent segments showed more variation, e.g. function words. Data-driven methods will model frequently occurring segments better. This might be an advantage, as frequent words will have a larger influence on the WER. Besides, ASR is based on statistics, and the differences and similarities perceived by humans might not be the most useful for ASR. A combination is thus usual, using some kind of data-driven method to verify the rules and find probabilities for them.

5.1 Examples of Knowledge Based Pronunciation Modeling

5.1.1 Linguistically Derived Phonological Rules

In [22], Tajchman et al. investigated ten phonological rules for US English. These ten rules are given in Table 2. We observe that the seven reduction rules have no context, whereas the three substitution rules have phone groups as context. Context given as phonetic features or phone groups is common for knowledge based phonological rules. A surface lexicon was built containing the unmodified pronunciations as well as the pronunciations resulting from the application of each relevant phonological rule. Each pronunciation was tagged with the use of rules in order to facilitate counting occurrences of each rule. The main issue in this paper was to estimate the pronunciation probabilities for unseen words using the rule probabilities. The rule probabilities found using automatic transcription (on WSJ) were similar to probabilities found using hand-labeled data on another set (TIMIT). This indicates that the phonological rules have a probability independent of the specific corpus. It was shown that pruning the resulting surface lexicon gave improved recognition results.

Probabilities for linguistically based transformation rules can also be determined by decision trees, as shown by Finken and Waibel [23]. Forced alignment was used to choose among the rules in the training corpus, which was the

Switchboard database (using a speaker adapted recognizer). These data-verified rules were modeled in a decision tree, taking into account the context dependency of phonetic neighbors, word type, speaking rate, average word or phone duration, vowel stress, pitch, and computed probabilities. A rule probability was estimated from the relative frequency of the use of each rule. The resulting transformation rules were interpreted as speaking mode dependent. The lexicon was expanded using pronunciations found by forced alignment, i.e. direct modeling, but with indirect modeling as an intermediate step. The variants found using the derived rules did not increase the performance as much as when selecting variants from the baseline dictionary. The authors interpreted this as due to added confusability. Pronunciation weighting using the speaking mode dependent decision trees increased the performance. The starting point for this experiment was knowledge based, but data were used for verification.

Five known pronunciation rules for Dutch (4 deletion rules and 1 insertion rule) were investigated by Kessens et al. in [24] and Wester et al. in [25]. Improvements were shown by incorporating them in known contexts. Modifying the acoustic models by retranscribing the training data using the variants gave increased improvement. Language model modification was done by incorporating pronunciation probabilities (computed from forced alignment of training data) and gave further improvement. Crossword rules were investigated by including both "border" versions of pronunciations and multi-words. In both cases this was limited to frequently occurring variations. A data-driven approach was compared with this knowledge-based approach by Kessens et al. in [26]. Deletion rules were found by allowing deletions in an alternative transcription, in order to let the acoustic models decide where to delete phones. The knowledge based approach and the data-driven approach gave about the same performance, but the data-driven approach resulted in a smaller lexicon. As the data-driven rule derivation was controlled by frequency counts, the most frequently occurring variations (the most important ones) were favored. There was a 96 % overlap in transcriptions by the two approaches. The data-driven rule context was phone identity, whereas the knowledge rule context contained broader groups of phones. The knowledge-based rules will therefore be applied more often, even if the same transformation (in this case deletion) is described.

5.1.2 Manually Transcribed Data

Using hand-labeled data is also a kind of knowledge-based method, but if the pronunciations found are used in retranscriptions, the categoriz-

ing is less clear-cut. One example of this is experiments performed during the Johns Hopkins summer workshop in 1997 described by Byrne et al. in [27] and Riley et al. in [28]. The results show that pronunciation modeling techniques using automatically labeled data performed better than hand-labeled. The experiments first used indirect pronunciation modeling using the rules to generate variants for unseen words. Decision trees were used to model the phonological rules seen in hand-labeled speech material, i.e. which phones can be neighbors dependent on lexical stress and distance from word boundary. From these decision trees a small network of possible alternate transcription was made for each word. Using all of these alternative pronunciations for recognition decreased the performance. This is the same effect as was shown in [23] and shows the need for some way to choose which pronunciations to accept in the lexicon. The alternative pronunciations from the decision trees were then used in a forced alignment to retranscribe the corpus. Using new decision trees based on this automatic transcription the performance increased. The authors' interpretation was that there was a mismatch between transcription by human perception and machine perception. Another reason may be that the hand-labeled material was much smaller than the automatically transcribed since the process of hand labeling is time consuming and expensive.

Direct modeling with hand-labeled data as bootstrap was also investigated in the experiments of the Johns Hopkins summer workshop in 1997 using an "explicit dictionary expansion", [27] and [28]. Pronunciations found sufficiently often in the hand-labeled or the automatically transcribed corpus were put in the lexicon with weights based on relative frequency. This means fewer variations for the recognizer, and the performance increased, as expected. Multi-words were used to model coarticulation effects be-

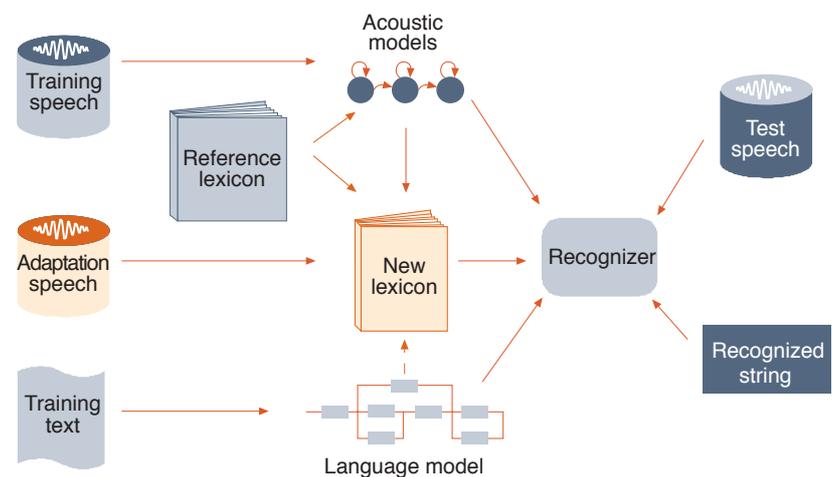


Figure 4 Recognition system with adapted lexicon

tween some of the words. The conclusion of the experiments was that crossword context is not necessary except for some words. Using the initial decision tree rules to retranscribe the training set and train new acoustic models increased the performance. These new models were used for a new retranscription, making new decision trees and new explicit lexicon expansions, improved the performance even more. These experiments also showed the need for care when deriving the weights for the alternate pronunciations (i.e. pronunciation probabilities).

5.2 Direct Data-driven Pronunciation Modeling

One of the major challenges in pronunciation modeling is to decide which pronunciations to include in the lexicon to get the best ASR performance. Most ASR lexica in use today are based on linguistic knowledge only and are not optimized with respect to ASR performance. Data-driven pronunciation modeling focuses on finding the “best” pronunciations given an objective criterion.

An overview of the recognition system with adapted lexicon based on data-driven methods is shown in Figure 4. The new lexicon is affected by both the adaptation data and the acoustic models, and for most approaches also the knowledge based reference lexicon. For consistency, the language model should also be considered in lexicon adaptation (but often it is not); this is shown by a dashed arrow. The acoustic models may also be adapted by the new data, possibly using the new lexicon. The new lexicon may also influence the language model. For an optimal system the influence between all parts of the system should be considered giving a joint optimization.

Pronunciation variation modeling can be described in two steps:

- 1 Find pronunciation variants
- 2 Assess the variants and modify the lexicon

5.3 Examples of Direct Data-driven Pronunciation Modeling

A truly data-driven method was reported by Holter and Svendsen in [29], using some of the methods derived for finding optimal pronunciations presented in [30]. For the experiments a US English database with a 991 word vocabulary was used. No rules or hand-labeled data were used, only a baseline recognizer. This recognizer was used to make an N -best list of pronunciations without any prior knowledge of the vocabulary other than the number of words and the boundaries of each word, which is usually present in an orthographic transcription. A subset of these pronunciations was chosen through a

maximum likelihood algorithm doing joint optimization for all the utterances of each word. A clustering procedure chose which variants to add in the lexicon to ensure distinct pronunciations, still using the maximum likelihood metric. The appeal of this idea is that it uses objective criteria for the optimization of all parts of the recognizer. The same method, using a 10-best phone loop, has been shown to give improvements for pronunciation modeling of Norwegian natural numbers [31].

Another approach to finding variants using data-driven methods was presented by Fosler-Lussier in [19] and [32]. First, an alternative transcription based on a bigram phone grammar was used to derive rules using decision trees. Then the training set was retranscribed using these rules to get a “smoothed transcription”, and the pronunciation variants that occurred frequently enough were added to the lexicon. An added feature in this approach is the use of “dynamic” lexica, including word frequency, word trigram probability, word length, and speaking rate measures in the pronunciation modeling.

A similar approach was described for the Verbomobil project by Wolff et al. in [33]. A phone recognizer generated alternative hypotheses that were aligned with a transcription based on the canonical lexicon entries. Pronunciations occurring eight times or more were included in the lexicon and gave an increased performance. A measure of confusability called “consolidation” showed that words occurring 15 times or more had reached a “stable” set of pronunciations. This threshold excluded 85 % of the lexicon adaptation material. The algorithm was therefore expanded to incorporate generalization by using frequent sequences instead of words only.

5.4 Indirect Data-driven Pronunciation modeling

Generating pronunciation variants using rules that are automatically derived from data is another modeling option. These rules should ideally capture the difference between the reference pronunciation of a word and the actual pronunciation used by the speakers. This approach is similar to the first step in the variant generation of [19] and [33], as well as the approach based on hand-labeled transcriptions in [28].

Decision tree modeling, also called CART (Classification and regression trees) modeling is a popular method for deriving pronunciation rules. It is described in some of the earliest pronunciation modeling approaches, e.g. [3] and [34]. Usually one tree is built for each phone. The number of rules is controlled by limiting the number of mappings in each leaf, and the leaf probabilities can be used as rule probabilities.

Pronunciation variant generation by using data-driven rule derivation can be described in five steps:

- 1 Automatically generate alternative transcriptions
- 2 Align the reference and alternative transcriptions
- 3 Derive rules from the alignment
- 4 Assess and prune the rules
- 5 Generate pronunciation variants from the rules, assess the variants, prune or assign weights, and modify the lexicon

As we can see the first step in direct pronunciation variant modeling is replaced by a 4-step rule derivation and a step 5 to generate variants from the rules. Step 4 is not trivial; rules may interact and one rule may change the context affecting other rules. The rule pruning will control the number of variants indirectly, but we need a step 5 to assess the variants. We may also add a step 6 performing retranscription of the lexicon adaptation material and iterate the process. The five steps are discussed more carefully below:

Step 1

The first step in rule generation is finding an alternative transcription that can reveal the true pronunciations of the speakers. We can also use knowledge as a starting point: if we have hand-labeled data, pronunciation rules can be derived from comparing this transcription with the reference. The reference transcription will also often be derived semi-automatically. Usually only a word transcription exists, and if the reference lexicon contains several pronunciations, the recognizer is used to choose pronunciation by forced alignment. The third possibility is to use a phone recognizer. As an example, the two transcriptions for the utterance “Paramount Pictures expected eight ...” are shown in Table 3.

Step 2

The usual approach when aligning the two transcriptions is to use dynamic programming. The difference between the methods lies in how the costs for the phone-to-phone mappings are as-

signed: often either uniform cost or phonetically based costs are used. The latter rely on knowledge about phone similarity and the assumption that the probability of phone-to-phone mappings due to pronunciation variation will follow phone similarities. In [35], an alternative alignment procedure was proposed using an estimated relation measure between the phones in the reference transcription and in the alternative transcription of the new speaker data. This measure utilizes statistically significant correspondence between the phones in the two transcriptions and was called association strength. An example of an alignment of the transcriptions in Table 3 is shown in Table 4.

Step 3

Rules representing the pronunciation variation can be extracted from the alignment of the two transcriptions. A usual approach in rule based pronunciation modeling is to let the rules express phone-to-phone mappings (allowing deletions and insertions). The rules are usually defined as dependent on a specified context. The width of the context has to be decided as well as which other information to include. The most frequently used context is one phone neighbour to each side of the phone(s) affected by the rule. This is different compared with knowledge-based rules where the context often is given as phonetic features as shown in Table 2. Also for data-driven rules other contexts than phone identity are used and contexts shown to have effect include word frequency [19], lexical stress, and syllabic information [28]. Using more complex contexts demands either more data to estimate the rules properly or a generalization of the contexts. CART trees can be used to generalize the context of a rule automatically. For crossword rules the word boundary information must be included in the context. From the example transcriptions for “Paramount Pictures ...” we can derive several rules. Examples of word internal phone-to-phone-mappings with context given by phone identity are:

[r ah m] maps to [r m] (a deletion)

Reference transcription (canonical pronunciations):

[p eh r ah m aw n t p ih k ch er z ih k s p eh k t ah d ey t]

Alternative transcription (phone loop on spontaneous dictation):

[p eh r m aa m p ih k ch er z ah k s p eh k t ih t ey iy t]

p eh r ah m aw n t p ih k ch er z ih k s p eh k t ah d ey t

p eh r m aa m p ih k ch er z ah k s p eh k t ih t ey iy t

Table 3 Example of reference and alternative phonetic transcriptions for the utterance “Paramount Pictures expected eight ...”

Table 4 Example of alignment of the transcriptions for the utterance “Paramount Pictures expected eight ...”

and

[m aw n] maps to [m aa m] (two substitutions)

In the last case a word-external rule may be more appropriate (\$ marks the word border):

[m aw n t \$ p] maps to [m aa m \$ p]

Step 4

Most rule-based pronunciation modeling techniques need some kind of pruning to control which of the alternative pronunciations that should be included in the lexicon. Rarely used pronunciations may introduce more errors than they correct. We can use a threshold based either on the rule probabilities, the pronunciation probabilities, or both, to control the number of new pronunciations to add.

Step 5

From the resulting set of selected rules new pronunciations are derived. Using several rules for each pronunciation will result in a huge number of new pronunciations:

$$\begin{aligned} \# \text{ pronunciations} = \\ (\# \text{ rules for phone 1}) \cdot (\# \text{ rules for phone 2}) \dots (6) \end{aligned}$$

The resulting pronunciation probabilities derived from the rule probabilities would be very low for most of the multi-rule derived pronunciations. One way around this is to use estimated pronunciation probabilities as a threshold to limit the number of variants instead of, or in addition to, the rule probability threshold.

Pronunciation probabilities can be estimated directly by counting the different pronunciations chosen when retranscribing the adaptation data by forced alignment (restricted by the initial rules). This was described as “smoothed transcription” in [19] and “explicit” dictionary expansion in [28]. Estimating pronunciation probabilities is not trivial for unseen words. Rule probabilities make it possible to estimate the probability for unseen words and pronunciations, but care must be taken. In [22], the problem of combining the rule probabilities to word probabilities was discussed. Pronunciation probabilities derived from the decision trees were compared with pronunciation probabilities estimated by frequency counts in [28]. These two ways of estimating the pronunciation probability did not give the same result.

When we have little data to derive pronunciations from, it may also be necessary to merge the reference lexicon and the new pronunciations [36]. In [19], merging was found beneficial even if a 100-hour adaptation set was used, because

the pronunciations were modeled directly and the infrequently occurring words got too “noisy” pronunciations.

5.5 Examples of Indirect Data-driven Pronunciation Modeling

Humphries and Woodland did experiments on British English accent modeling in [37]. This is an example of indirect modeling, deriving pronunciation rules from data. No hand-labeled transcription was used, so this is an entirely data-driven approach. Alternate transcriptions were found by allowing all vowels to be substituted and then using a forced alignment. Vowel transformations are an important difference between British English accents. The three best transcriptions were used to derive rules using context dependent decision trees including leaf probabilities. A new pronunciation dictionary was then made for the new accent. Typically, an average of 4 pronunciations per word were found to be effective. The test was performed on a 2000 word vocabulary, and adding pronunciations increased the performance.

Humphries and Woodland have also done experiments on British versus US English in [36] and [38]. The recognizer was here used to perform a free (and erroneous) transcription. An acoustic confidence measure was used to filter the transcribed data before rule derivation. This transcription was aligned with the canonical pronunciations in the British English lexicon. A list of possible phone transformations (substitutions, insertions and deletions) was generated and incorporated in a decision tree. Examination revealed interesting correlations with linguistic analysis of the differences between British and US English, e.g. transformation of [t] to [d] in certain contexts. The British-trained recognizer was tested on US English speech using US-adapted pronunciations, and this gave an increased performance compared with using British pronunciations [36]. There was no difference between using British pronunciations and adapted US pronunciations when training new acoustic models on US English speech [38]. The performance was worse but comparable to US models trained on “real” US pronunciations. In the case of sufficient training material, the extra phonological information was of less value. The authors suggested that in this case the pronunciation variation was taken care of by the acoustic models.

An example of the migration from knowledge based to data-driven modeling, is the work of Cremelie et al. Their first work was based on hand-labeled data [39]. Later work in [40] and [41] was based on automatically derived “expert” transcriptions where a constrained speech recognizer was used. In the improved

version [40], more linguistic constraints were put into the automatic transcription, a wider context was used, and negative rules were allowed. Negative rules means that variation is prohibited. In [41], further improvements gave a significant performance gain. Pronunciation rules were derived from alignment of the reference and “expert” transcriptions, and each acoustic segment was only allowed to count for one rule. Rule probabilities were found by frequency counts and the rules were ordered in a rule hierarchy that favored more specific rules. The most important rules were the coarticulation rules between words; most other experiments only consider intra-word variations. These coarticulation rules were incorporated in the language model (since cross-word rules cannot simply be added to the lexicon). For the experiments, they used two databases with different vocabularies for both English and Flemish, i.e. four databases in total. Cross-checking between the two databases for the same language was done to investigate possible corpus-specific rules. The results showed improvement for both languages and about the same level of improvement for rules based on either automatically generated or hand-made transcriptions. The automatic transcription version was best in some tests, the same effect as observed by Byrne et al. [27]. According to the authors, the reason could be that the automatic transcription caught the peculiarities of the recognizer and therefore gave rules better suited for this particular recognizer.

In [42], Amdal et al. used data-driven approaches for all the steps in indirect data-driven pronunciation modeling. For the alignment of the reference and alternative pronunciations the *association strength* introduced in [35] was used to derive phone substitution costs from the data. A metric based on acoustic log likelihood was used in rule pruning. The methods were evaluated on a non-native task. The results showed that the acoustic log likelihood pruning improved the ASR performance compared with the more traditional rule probability pruning. A better performance was observed when modeling the non-native speakers jointly than individually. This was a surprising result, as the speakers had quite different language backgrounds, but may be caused by the small amount of data available. Even if the joint set of non-native speech was more diverse, the larger amount of data was beneficial to get a more reliable rule selection for the data-driven methods investigated.

5.6 Confusability Reduction

Confusability reduction is closely linked to the pruning of rules and pronunciation variants. As shown the assessment of pronunciations is usually done by one or a combination of these methods:

- Assess pronunciation rules (the resulting variants will then be assessed indirectly)
- Assess each pronunciation variant directly

Modeling pronunciation variation by adding several pronunciations for each word in the lexicon should be done with care. More variants, and probably more similar variants, will increase the lexical confusability and the error rate. A method of balancing the “old” errors corrected and the “new” errors introduced should be included in the algorithm. Rule and variant assessment only considers the “goodness” of each pronunciation alone and how it performs on the word that the pronunciation belongs to. In general, the pronunciations will interact; thus, a more global approach to assess the total set of pronunciations that takes into account the effect on the other words in the vocabulary should be beneficial. This calls for discriminative techniques incorporating misclassification measures.

Torre et al. explored in [43] measures of word confusability by first estimating a phone confusion matrix. The phone confusions were combined with word confusions that were used in vocabulary selection. If we have several synonyms, this algorithm can be used to choose the best pronunciations among synonyms. With basis in the same algorithms, automatic alternative transcription generation was also examined with promising preliminary tests. Phone confusion matrices were also used by Sloboda and Waibel in [44]. First the confusion matrix was used to smooth a phone bigram that was used to automatically find variants for frequently misrecognized words. For the variants found, homophones as well as variants that only differed in confusing phones, were eliminated.

Discriminative model combination was used for pronunciation modeling experiments by Schramm and Beyerlein in [45]. An expression for the word error count including the pronunciation weights was derived. Minimizing this function with respect to these weights gave an iteration formula for updating the weights. The update function depends on the frequency of occurrence of the pronunciation in the true word as well as for competing words. The technique presented is similar to the one presented by Korkmazskiy and Juang in [46], where discriminative adaptation of pronunciation weights were used.

A confusability measure based on substring matching was presented by Wester and Fosler-Lussier in [47]. All possible word pronunciations that matched substrings in the reference transcription were used to make a lattice of confusable words. The confusability metric was calculated by considering the number of words that

corresponded to each phone. Pronunciations with high confusion counts were then removed.

This method was later extended to incorporate acoustic confusability by Fosler-Lussier et al. in [48]. The FST formulation of the ASR system as shown in equation (3) is utilized by inverting the FSTs. A weighted set of all word sequences \mathbf{W} confusable with any word sequence W was derived by composing the given word sequence with inverted transducers until acoustic scores were produced, and then reversing the process:

$$\mathbf{W} = W \circ L \circ P^{-1} \circ T^{-1} \circ T \circ P \circ L \quad (7)$$

A confusion matrix represented by a FST C was used to model $T^{-1} \circ T$. This confusion matrix was trained on the recognition errors of a training set. It was shown that the confusion matrix formalism was able to reasonably predict recognition errors in the test set at least better than chance.

6 Concluding Remarks

We have given an overview of pronunciation modeling by lexicon adaptation. There has been a migration from knowledge-based methods to more data-driven methods and combinations of the two.

A use of pronunciation modeling not covered in this article is computer-assisted language learning (CALL) systems. In second-language instruction technologies, pronunciation variation modeling can help capture regular patterns of errors made by students. One challenge in this domain is that the pronunciations of the users (hopefully) will change over time.

For most experiments, the improvements seen are modest. A general observation is that adding variants to the ASR lexicon not only corrects errors, but also introduces errors. One reason for the modest improvements achieved in pronunciation modeling is the lack of a way to control the confusability between pronunciations. To make lexica tailored to a person or group we cannot rely on just adding extra pronunciations, we must also remove confusable ones. One of the main topics in pronunciation variation modeling presently is therefore confusability measures. Speaker dependent lexicon adaptation is also an interesting subject in the current focus of the pronunciation modeling community.

7 References

- 1 Weintraub, M et al. Effect on speaking style on LVCSR performance. In: *Proc. ICSLP-96*, Philadelphia (PA), USA, Addendum 16–19, 1996.

- 2 Bahl, L R et al. Large vocabulary natural language continuous speech recognition. In: *Proc. ICASSP-89*, Glasgow, Scotland, 465–467, 1989.
- 3 Riley, M D, Ljolje, A. *Automatic speech and speaker recognition: Advanced topics*, ch. Automatic generation of detailed pronunciation lexicons, 285–301. Kluwer, 1996.
- 4 Strik, H, Cucchiari, C. Modeling pronunciation variation for ASR : overview and comparison of methods. In: *Proc. ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, Rolduc, the Netherlands, 137–144, 1998.
- 5 Strik, H. Pronunciation adaptation at the lexical level. In: *Proc. ISCA ITRW Adaptation methods for speech recognition*, Sophia-Antipolis, France, 123–130, 2001.
- 6 Adda-Decker, M. Towards multilingual interoperability in automatic speech recognition. *Speech Communication*, 35 (1-2), 5–20, 2001.
- 7 Mohri, M et al. Full expansion of context-dependent networks in large vocabulary speech recognition. In: *Proc. ICASSP-98*, Seattle (WA), USA, 665–668, 1998.
- 8 Mou, X, Seneff, S, Zue, V. Context-dependent probabilistic hierarchical sub-lexical modelling using finite state transducers. In: *Proc. EUROSPEECH-2001*, Aalborg, Denmark, 451–455, 2001.
- 9 Jurafsky, D et al. What kind of pronunciation variation is hard for triphones to model? In: *Proc. ICASSP-2001*, Salt Lake City (UT), USA, 577–580, 2001.
- 10 Van Compernelle, D. Recognizing speech of goats, wolves, sheep and ... non-natives. *Speech Communication*, 35 (1-2), 71–79, 2001.
- 11 Stolcke, A, Shriberg, E. Statistical language modeling for speech disfluencies. In: *Proc. ICASSP-96*, Atlanta (GA), USA, 405–408, 1996.
- 12 *CMU Pronunciation Dictionary*. [online], 1998. [cited 2002-03-01]. URL: <http://www.speech.cs.cmu.edu/cgi-bin/cmudict/>
- 13 Lea, W A. *Trends in speech recognition*. NJ, Englewood Cliffs, Prentice Hall, 1980.

- 14 Greenberg, S, Hollenback, J, Ellis, D. Insights into spoken language gleaned from phonetic transcription of the Switchboard corpus. In: *Proc. ICSLP-96*, Philadelphia (PA), USA, Addendum 24–27, 1996.
- 15 McAllaster, D et al. Fabricating conversational speech data with acoustic models : A program to examine model-data mismatch. In: *Proc. ICSLP-98*, Sydney, Australia, 1847–1850, 1998.
- 16 Amdal, I, Svendsen, T. Evaluation of pronunciation variants in the ASR lexicon for different speaking styles. In: *Proc. LREC-2002*, Las Palmas de Gran Canaria, Spain, 1290–1295, 2002.
- 17 Livescu, K, Glass, J. Lexical modeling of non-native speech for automatic speech recognition. In: *Proc. ICASSP-2000*, Istanbul, Turkey, 1683–1686, 2000.
- 18 Giachin, E P, Rosenberg, A E, Lee, C-H. Word juncture modeling using phonological rules for HMM-based continuous speech recognition. *Computer Speech and Language*, 5 (2), 155–168, 1991.
- 19 Fosler-Lussier, E, Williams, G. Not just what, but also when: Guided automatic modeling of Broadcast News. In: *Proc. DARPA Broadcast News Workshop*, Herndon (VA), USA, 171–174, 1999.
- 20 Fosler-Lussier, E, Morgan, N. Effects of speaking rate and word frequency on pronunciations in conversational speech. *Speech Communication*, 29 (2-4), 137–158, 1999.
- 21 *CALLHOME American English Lexicon (PRONLEX)*. [online description], 1995. [cited 2002-03-01]. URL: <http://morph ldc.upenn.edu/Catalog/LDC97L20.html>.
- 22 Tajchman, G, Fosler, E, Jurafsky, D. Building multiple pronunciation models for novel words using exploratory computational phonology. In: *Proc. EUROSPEECH-95*, Madrid, Spain, 2247–2250, 1995.
- 23 Finke, M, Waibel, A. Speaking mode dependent pronunciation modelling in large vocabulary conversational speech recognition. In: *Proc. EUROSPEECH-97*, Rhodes, Greece, 2379–2382, 1997.
- 24 Kessens, J M, Wester, M, Strik, H. Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation. *Speech Communication*, 29 (2-4), 193–207, 1999.
- 25 Wester, M, Kessens, J M, Strik, H. Pronunciation variation in ASR : Which variation to model? In: *Proc. ICSLP-2000*, Beijing, China, IV:488–491, 2000.
- 26 Kessens, J M, Strik, H, Cucchiari, C. A bottom-up method for obtaining information about pronunciation variation. In: *Proc. ICSLP-2000*, Beijing, China, I:274–277, 2000.
- 27 Byrne, W J et al. Pronunciation modelling using a hand-labelled corpus for conversational speech recognition. In: *Proc. ICASSP-98*, Seattle (WA), USA, 313–316, 1998.
- 28 Riley, M et al. Stochastic pronunciation modelling from hand-labelled phonetic corpora. *Speech Communication*, 29 (2-4), 209–224, 1999.
- 29 Holter, T, Svendsen, T. Maximum likelihood modelling of pronunciation variation. *Speech Communication*, 29 (2-4), 177–191, 1999.
- 30 Holter, T, Svendsen, T. Incorporating linguistic knowledge and automatic baseform generation in acoustic subword unit based speech recognition. In: *Proc. EURO-SPEECH-97*, Rhodes, Greece, 1159–1162, 1997.
- 31 Amdal, I, Holter, T, Svendsen, T. Maximum likelihood pronunciation modelling of Norwegian natural numbers for automatic speech recognition. In: *Proc. Norwegian Signal Processing Symposium (NORSIG)*, Asker, Norway, 145–150, 1999.
- 32 Fosler-Lussier, E. Multi-level decision trees for static and dynamic pronunciation models. In: *Proc. EUROSPEECH-99*, Budapest, Hungary, 463–466, 1999.
- 33 Wolff, M, Eichner, M, Hoffmann, R. Automatic learning and optimization of pronunciation dictionaries. In: *Proc. ISCA ITRW Adaptation methods for speech recognition*, Sophia-Antipolis, France, 159–162, 2001.
- 34 Bahl, L R et al. Decision trees for phonological rules in continuous speech. In: *Proc. ICASSP-91*, Toronto, Canada, 185–188, 1991.
- 35 Amdal, I, Korkmazskiy, F, Surendran, A C. Data-driven pronunciation modelling for non-native speakers using association strength between phones. In: *Proc. ISCA ITRW ASR2000*, Paris, France, 85–90, 2000.

- 36 Humphries, J J, Woodland, P C. Using accent-specific pronunciation modelling for improved large vocabulary continuous speech recognition. In: *Proc. EURO-SPEECH-97*, Rhodes, Greece, 2367–2370, 1997.
- 37 Humphries, J J, Woodland, P C, Pearce, D. Using accent-specific pronunciation modelling for robust speech recognition. In: *Proc. ICSLP-96*, Philadelphia (PA), USA, 2324–2327, 1996.
- 38 Humphries, J J, Woodland, P C. The use of accent-specific pronunciation dictionaries in acoustic model training. In: *Proc. ICASSP-98*, Seattle (WA), USA, 317–320, 1998.
- 39 Cremelie, N, Martens, J-P. Automatic rule-based generation of word pronunciation networks. In: *Proc. EUROSPEECH-97*, Rhodes, Greece, 2459–2462, 1997.
- 40 Cremelie, N, Martens, J-P. In search of better pronunciation models for speech recognition. *Speech Communication*, 29 (2-4), 115–136, 1999.
- 41 Yang, Q, Martens, J-P. Data-driven lexical modeling of pronunciation variations for ASR. In: *Proc. ICSLP-2000*, Beijing, China, I:417–420, 2000.
- 42 Amdal, I, Korkmazskiy, F, Surendran, A C. Joint pronunciation modelling of non-native speakers using data-driven methods. In: *Proc. ICSLP-2000*, Beijing, China, III:622–625, 2000.
- 43 Torre, D et al. Automatic alternative transcription generation and vocabulary selection for flexible word recognizers. In: *Proc. ICASSP-97*, Munich, Germany, 1463–1466, 1997.
- 44 Sloboda, T, Waibel, A. Dictionary learning for spontaneous speech recognition. In: *Proc. ICSLP-96*, Philadelphia (PA), USA, 2328–2331, 1996.
- 45 Schramm, H, Beyerlein, P. Towards discriminative lexicon optimization. In: *Proc. EUROSPEECH-2001*, Aalborg, Denmark, 1457–1460, 2001.
- 46 Korkmazskiy, F, Juang, B-H. Discriminative training of the pronunciation networks. In: *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara (CA), USA, 137–144, 1997.
- 47 Wester, M, Fosler-Lussier, E. A comparison of data-derived and knowledge-based modeling of pronunciation variation. In: *Proc. ICSLP-2000*, Beijing, China, I:270–273, 2000.
- 48 Fosler-Lussier, E, Amdal, I, Kuo, H-K J. On the road to improved lexical confusability metrics. In: *Proc. ISCA ITRW on Pronunciation Modeling and Lexicon Adaptation (PMLA)*, Estes Park (CO), USA, 53–58, 2002.

The 'Melody' in Synthetic Speech: What Kind of Phonetic Knowledge Can Be Used to Improve It?

J Ø R N A L M B E R G



Jørn AlMBERG (32) is finalising his doctoral thesis in phonetics by the autumn of 2003 at the Department of Language- and Communication Studies at the Norwegian University of Science and Technology (NTNU). His thesis will present a quantitative acoustic analysis of intonation produced in different Norwegian dialect regions. The study uses data from the Telenor speech database TABU.0. AlMBERG's research interests are mainly experimental-acoustic studies of prosody in Norwegian speech, and often with a dialectal perspective. He has participated in a number of interdisciplinary projects along with the Department of Telecommunications at NTNU and/or the Speech technology group at Telenor R&D. He has also developed some web-based speech resources, e.g.

<http://www.ling.hf.ntnu.no/nos/>

jorn.almberg@hf.ntnu.no

Speech melody, or intonation, is a phenomenon that is of interest to both speech engineers and phoneticians. We know that intonation plays a central part when it comes to the perceived naturalness of synthetic speech. We also know that continuous synthesized speech is much easier to understand if it has "good" intonation.

"What is good intonation?" is a crucial question for a speech engineer. "What is a typical intonation?" might be the corresponding question for a phonetician. The answers to these two questions are hardly very different from each other. On the contrary, the speech engineer and the phonetician often engage in the same problems. An example of the close relationship between the two perspectives is that phoneticians often use speech synthesis in their perceptually oriented intonation studies. But the phonetician and the speech engineer usually choose different methods in their research of the same object. Consequently their results are also of different natures: A phonetician often describes intonation through phonological features like *High*, *Low*, *Rise* and *Fall*, while a speech engineer prefers results represented as numerical values based on statistical analyses.

This article will look at the methodological approach of a number of phonetic studies on Norwegian tone and intonation. Furthermore, it will present preliminary findings from the author's ongoing phonetic study of Norwegian intonation, in order to suggest how a phonetic intonation study based on linguistic analysis may be of more immediate value to speech technology research and development.

1 Background

Spoken language is the primary object of study for many linguistic disciplines in addition to phonetics. For a phonetician it is natural to split speech research into two groups: *segmental* and *prosodic* studies. *Segmental* studies focus on the different speech sounds. An example of a segmental study is a diachronic survey of the pronunciation of the phoneme /r/ in Norwegian (e.g. Foldvik 1977). *Prosodic* studies focus on larger entities of speech, e.g. syllables, words or utterances.

The most studied prosodic property in spoken Norwegian is what has often been called *lexical tone*. In short, most dialects of spoken Norwegian must generate one of two possible "melodies" to a word when uttered in isolation. Norwegian can thus distinguish between pairs of words based on their melody. An example of this is the word pair "Hammer" (which is a common Norwegian surname) [1ham:er] and "hammer" (Eng. "hammer") [2ham:er]. Figures 1 and 2 illustrate the tonal difference between the two words.¹⁾

In addition to most dialects of Norwegian and Swedish, this tonal system called *lexical tone* or pitch accent is only found in some varieties of Serbo-Croat (Cruttenden 1997: 11). This exotic aspect is probably an important reason for the numerous amounts of Norwegian studies with focus on word tone, while there have been very few studies of intonation contours.

Norwegian phoneticians and linguists have often focused on the Norwegian *word tones* or *tonemes* in their studies, and some of these studies have also proved that tonality is an important dialect marker. All such tonal studies can be put somewhere along a scale ranging from impressionistic estimation to instrumental measuring. By impressionistic studies we mean work that is based on the researcher's own subjective estimations of the object analysed. An instrumental study is mainly based on acoustic measuring of the F0 contour.

The next section will present a few tonal studies of Norwegian, and discuss the methodological traditions that these have been founded on. We will look at studies of both *word tone* and *intonation*, and we will use "tonal studies" as a collective term for both these types of studies. We will look at details like choice of speech material, presentations of acoustic analyses and statistical treatments of the data from these analyses.

2 A Selection of Norwegian Tonal Studies

2.1 Ivar Alnæs; Musical Notes and Violins

If we look at studies of spoken Norwegian, it is hard to find works based on instrumental measuring earlier than the first published work by Ernst W. Selmer in 1917. Before 1917 the explorations that is today considered tonal studies of

¹⁾ Sound samples and graphic illustrations of the Norwegian word tones in different dialects can be found on this website from the internet resource Norske språkllyder (Eng. Norwegian speech sounds): http://www.ling.hf.ntnu.no/ipa/no/tema_008.html

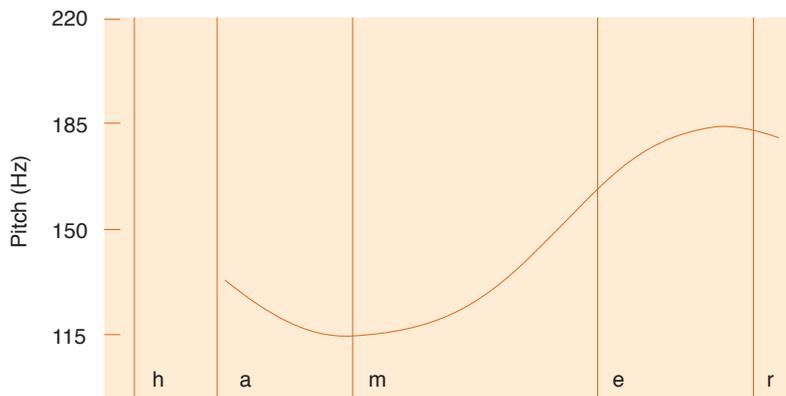


Figure 1 Pitch contour of an East Norwegian pronunciation of the Norwegian surname “Hammer” [¹ham:er]. Segment boundaries indicated by vertical bars

Figure 2 Pitch contour of an East Norwegian pronunciation of “hammer” (Eng. ‘hammer’) [²ham:er]. When comparing the contours in Figures 1 and 2, we see that the tonal difference between them is mainly located towards the left edge of the word

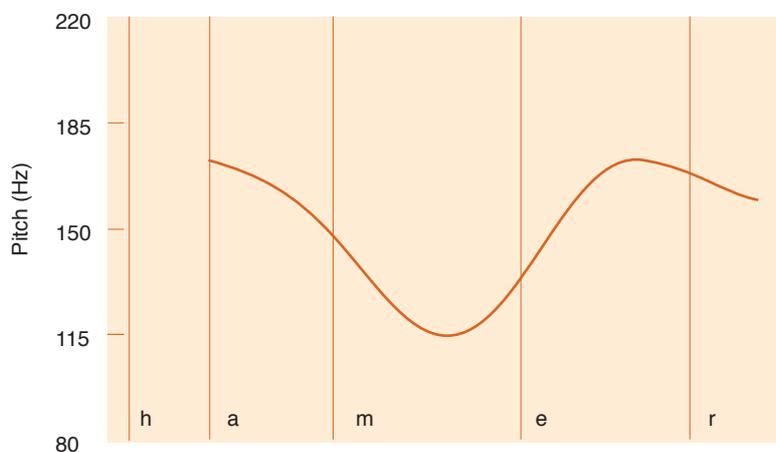


Figure 3 Tonal annotation by Alnæs (1916: 94). The example consists of two tonal groups: (Dæ’k’ saa) and (svært læng’ si’n jæ saa’n). A vertical bar separates the two tonal groups



most important instrument for analysis. Based on their auditive impressions they both tried to imitate tonal movements of speech with violins, so that they in the next turn could draw musical notes.

Figure 3 illustrates how Alnæs used musical notes in his tonal descriptions. It also shows Alnæs’ point that tonal groups can consist of more than one word: the two tonal groups in Figure 3 consist of four and five words, respectively.³⁾

Alnæs sometimes used a *phonograph*, and later a *parlograph*, which were sound recorders that were available at his time. The recorders made it possible to listen to the speech sample more than once, thus making Alnæs’ musical annotations more accurate than when he listened to “live speech”.

At this time (1916) instrumental measuring of tonal height had been performed for several decades internationally. Alnæs was aware of this, and regretted the fact that such equipment was not yet available in Norway. He comforted himself with the rumours saying that instrumental analyses were very time demanding, and that his own methodology made him much more flexible in space and time. “I have, so to say, my laboratory with me everywhere” (Alnæs 1916: 3; our translation). The instruments used in foreign studies at that time were not very flexible, as they were able to analyse only short recordings of lab-speech at a time.

2.2 Arne Vanvik – Introspection, and the Dialect Aspect

For reasons of space, we skip studies like Mo (e.g. 1925), Selmer (e.g. 1928), and Haugen and Joos (1952), and move on to the works on Norwegian word tone by Arne Vanvik. Vanvik wrote a book about the Trondheim dialect (Vanvik 1966) and published two articles, which described “standard East Norwegian” (Vanvik 1972 and 1973). These works describe the tonal

²⁾ The entity Alnæs called *tonelagsgruppe* in Norwegian has later been referred to with the terms *Foot* (Fretheim and Nilsen 1989), and *Accent Phrase* (Kristoffersen 2000: 275 pp).

³⁾ The orthography of the first tonal group reflects a number of elisions, which makes it look like only two or three words instead of the original four.

properties of these speech types, but generally he put very little effort in the description of tonality levels above the word level.

We will look at an early article by Vanvik, which deals with what seems to be a crucial point for him: a description of the tone in monosyllabic words (as well as polysyllabic words with stress on the last syllable) as a particular *third word tone*, that is neither *word tone 1* or *2* – a statement that is in opposition to most other tonal studies.

All of Vanvik's works were almost completely based on impressionistic analyses, a fact that is illustrated here: "I have not been able to on a perceptual basis find any approximately firm word tone in stressed monosyllabic words" (Vanvik 1956: 209; our translation from Norwegian), and "By listening to older people's speech I haven't gained the impression that east Norwegian intonation has changed radically" (Vanvik 1956: 210; our translation). Vanvik also described the tonality of the "dialects" spoken in the cities of Trondheim, Bergen, Oslo and Stavanger.

Figure 4 illustrates how Vanvik describes the tonal properties of the two word tones called *Aksent 1* and *Aksent 2* in the Stavanger dialect.

Vanvik calls his representations "auditory descriptions [that are] averages of the different ways the examples were pronounced". He uses "several people from Stavanger and Bergen" (Vanvik 1956: 213; our translation) as informants. As informant for Trondheim and Oslo he uses himself. The illustrations of the tonal accents are abstracted contours with solid lines for stressed syllables, and dotted lines for unstressed syllables. The contours are positioned between an upper and a lower horizontal line, which symbolise the tonal extremities.

In the final part of the article Vanvik explains how intonation may affect the "word tones". The examples he uses to document this are illustrations with the same form as shown in Figure 4. In this final part he uses himself as the only informant: "I will try to classify the occurrences in which I in my own East Norwegian have a falling and a rising intonation" (Vanvik 1956: 215; our translation).

2.3 Fintoft and Mjaavatn – Instrumental Analyses of Dialect Variation

In the 1980s Knut Fintoft and Per Egil Mjaavatn did a number of projects focussing on word tone. Fintoft and Mjaavatn (1980) investigated the production of tonal properties in different Norwegian dialect regions. The article suggests a partition into dialect regions based on difference



Figure 4 Tonal annotation of the Stavanger dialect by Vanvik (1956: 97), grouped on the basis of number of syllables (Norwegian: stavelser)

in phonetic realisation of the tonal accents. The authors base their proposition on analyses of speech recordings of about 1000 adult informants from about 450 different locations in Norway. The speech material consists of about 40 disyllabic word pairs with differing tonal accent, all read within the carrier phrase *Det var ... du sa* (Eng. "It was ... you said") (Fintoft et al. 1980: 67). The word list was adjusted to the vocabulary for the different dialects. Informants from Mid- and North Norway were asked to adjust their pronunciation towards the written text by pronouncing some of the words on the list closer to "Bokmål" than they would normally do. The recordings were analysed with a pitch meter, which gave them fundamental frequency (F0) values in Hertz. Both F0 and time values were normalised (Figure 5).

Fintoft and Mjaavatn used seven *measuring points* in their analyses of single words. Based on statistical treatment of these points, they ended up with combinations of four different "typical" tonal curves, referring to the four dialect regions *North Norwegian*, *Trønder*, *West Norwegian* and *East Norwegian*.

Fintoft and Mjaavatn's conclusion is that the combination of how the two tonal accents are realised should be used extensively as an isogloss in Norwegian dialectology, as a separator of both regional and of local variants of spoken Norwegian.

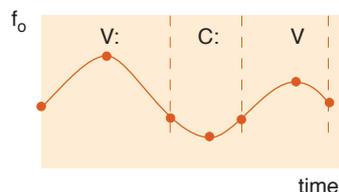


Figure 5 Seven measuring points, as illustrated in Fintoft et al. (1980: 68)

2.4 Fretheim and Nilsen – The Trondheim Model

The *Trondheim Model* is the name of a project that was initialised at the end of the 1970s, and that has resulted in, among other things, a doctoral dissertation from 1992 about the interaction between intonation and syntax in South-East Norwegian (Nilsen 1992). The *Trondheim Model* is a systematic description of South-East Norwegian intonation. The explanation for the name is that the model has been carried out at the University of Trondheim (later called NTNU), by Thorstein Fretheim and Randi A. Nilsen. We will look at two of their published works with reference to this model. The main aim of the model is not to describe the word tone, but to take a phonological perspective towards the *intonation* properties of spoken Norwegian, and more specifically how these properties coincide with pragmatic and syntactic mechanisms.

In Fretheim and Nilsen (1988) the two authors focus on the tonal properties of so-called alternative questions of the type *Is it a blue ball, or is it green?* This article has several illustrations in the form of F₀ contours based on instrumental analyses of speech recordings. Some of the illustrations include values in Hertz for certain phases that are considered to be the most crucial contours. In addition to South Eastern Norwegian, analyses of speech from other Norwegian regions are presented. The authors want to depict that different dialects use different tonal variants in order to give the same pragmatic signal to the listener(s). Some of the postulations in the article are based on pitch contours from speech recordings that are “reconstructions of an intonation pattern that were used by a person from Lillesand in a radio program [...]” (Fretheim et al. 1988: 96, our translation). This means that the authors in these cases have carried out instrumental analyses of their own imitations of dialects that are not their own.

Fretheim and Nilsen (1989) is another example of an intonation study with a dialectal perspective. As early as in the 19th century there were documented different “melodies” in different Norwegian dialects (e.g. Storm 1874, and later Alnæs 1916). We saw a more recent documentation of this fact in Fintoft and Mjaavatt (1980). Fretheim and Nilsen (1989) present their view of what are the tonal differences between *high tone* (Mid- and East Norwegian) and *low tone* (North- and West Norwegian) dialects, and end up with a description of the tonal properties in Romsdal in Western Norway; a region close to one of the borders between the *high tone* and *low tone*

dialects. They use terminology established in Haugen and Joos (1952) to describe high tone intonation as opposed to the low tone system. They pay much attention to the difference between high tone and low tone dialects’ phonetic realisations of *focus*,⁴⁾ and they illustrate this with pitch analyses of one utterance spoken with high- and low tone dialects, respectively. One informant from each of two appropriate dialect regions read the utterances that were analysed.

Methodologically, the most interesting aspect is found in the latter parts of the article, where Fretheim and Nilsen present the first known formalised perception study of tonal aspects in spoken Norwegian. The perception test aims to find what tonal movements – high tone or low tone – are considered most natural to speakers from the borderline region in Romsdal. Original and synthetically manipulated versions of three different utterances were used in a listening test. Four different intonation contours were created for each of the three utterances, based on how the authors meant they would have been pronounced along a scale where typical high- and low tone versions were the two extreme versions. In addition to their own language intuition, the authors based their creation of stimuli on studies of production data from the Romsdal dialect that was the object studied. Because, as they say: “People do not always talk the way they believe they do.” (Fretheim and Nilsen 1989: 448). (They do not present the data from the production study, though.) The stimuli were presented in a discrimination test, with 32 listeners from Møre and Romsdal (most of them from Romsdal) who were asked to tell which of the synthesised utterances within each pair resembled their own way of speaking the most. The results from the listening test are not very clear. The authors expected that the listeners would prefer one or both the intermediate versions of the utterances, but the results showed that for one utterance, as many as 14 of the 32 listeners preferred the high tone version to the intermediate ones. The authors claim that this is the result of an intonation modelling that was not optimal. Furthermore, it should be mentioned that none of the results are presented through statistical treatment, or with any kind of tables or graphics. All the results are presented and explained through plain text.

In their conclusion Fretheim and Nilsen say that the phonetic realisation of focus is the most crucial difference between these borderline dialects, as compared to neighbouring dialects being typical high- or low tone dialects. Before this study it was an established belief that all Norwegian

⁴⁾ *Focus* is a term used to group the tonal units with respect to how prominent they are compared to each other. The most prominent tonal units are focal, while less prominent tonal units are non-focal. The Trondheim model says that a grammatical Norwegian utterance has one or two focal units.

dialects have their focal peak at either the left- (high tone dialects) or right (low tone dialects) edge of a tonal unit. But Fretheim and Nilsen found that the focal peak in the borderline dialect they studied could occur sometimes at the left edge, and sometimes at the right edge of the tonal unit, all depending on the context. The authors also point out that their investigation has dealt with intonation properties of just one of at least three borderline regions that exist in Norway.

Two examples of more recent studies of Norwegian intonation are van Dommelen, Fretheim and Nilsen (1998), and Husby and Almberg (2002). For reasons of space, we will just give a brief and collective outline of these two. Both these studies include the perceptual perspective in the form of listening tests, and they present results from the listening tests in tables including both raw data and statistical analyses.

2.5 Discussion

The opposition between phonetics as an instrumental discipline and phonemics/phonology as an impressionistic discipline was blown off by Bertil Malmberg some 40 years ago (Malmberg 1962). Malmberg claims that a study by Jakobson and Halle (1952) represents the first serious attempt at combining impressionistic and instrumental analysis within the field of linguistics.⁵⁾ The approach of Jakobson et al. (1952) was a combination of structural linguistics and acoustics, and had its starting point in a milieu working with telecommunication, with a need for a basic knowledge of what language is, on its own (scientific) premises.

Malmberg (ibid.) claims that instrumental analyses that do not include the auditive aspect are of little value. Pure production analyses like Fintoft et al. (1980) will therefore not be very interesting in Malmberg's perspective. Of the studies we looked at above, only the three most recent studies; Fretheim et al. (1989), van Dommelen et al. (1998), and Husby et al. (2002), combined an auditive/impressionistic and an acoustic/instrumental approach.

Non-instrumental approaches like Vanvik (1956, 1972, 1973) should also be commented here. In a later article, Vanvik (1983) attacks the many instrumentally based studies presented in that period. Vanvik says that one should never be tempted to let the instruments replace the human ear when it comes to describing human speech. "We must go back to hearing as the primary tool

for phonetic studies" (Vanvik 1983: 1; our translation).

In Vanvik (1956) we saw a tonal study of four urban Norwegian dialects, where the analyses and descriptions were totally based on the author's own auditive impression. We also saw that Vanvik used himself as informant for *two* of the dialects involved (Oslo and Trondheim), which makes it reasonable to ask about the authenticity of the dialectal variants owned by the author. One may also query the author's ability to perceive and describe dialects other than his own in the way he does. When Vanvik excludes instrumental measurements in his studies, he makes the ear not only the *primary*, but also the *only* instrument. Vanvik has also been criticised for his unilateral focus on analyses of single word units. Studies that do not consider the fact that natural speech most often consists of a continuous string of a number of words, is of less interest and value, a point that was made about Norwegian tonal studies as early as 1916 by Alnæs. This criticism also goes to Fintoft et al. (1980), who analysed tonal units consisting of single words only.

3 Interplay: The Trondheim Model as a Description Tool

An instrumental analysis of speech has to be founded on some kind of auditory-based annotation of the speech material, carried out within a linguistic framework. If one wants to analyse e.g. the intonation of spoken Norwegian, the description tool developed through the work with the Trondheim model (e.g. Nilsen 1992) is a natural choice.⁶⁾ An example of such an annotation is shown in Figure 6, taken from Nilsen (1992).

The Trondheim model consists of a prosodic hierarchy, ranging from the syllable level and up to the utterance level. In Figure 6, the levels *Foot* (F), *Intonational Phrase* (IP) and *Intonational Utterance* (IU) are used. In addition, the illustrated annotation assigns the Foot units as either *focal* (UPPER CASE) or *non-focal* (lower case).

Below we take a closer look at the different levels in the Trondheim model.

3.1 Syllable

This is the smallest constituent in the Trondheim model. Syllables can be either strong (i.e. stressed) or weak (unstressed). Stressed syllables

⁵⁾ Malmberg considers phonetics to be a sub-branch within the field of linguistics.

⁶⁾ The Trondheim model has some documented limitations when it comes to compatibility to dialect regions that are intonationally different from the East Norwegian system (e.g. Abrahamsen, 1998). Typical "deviating" regions are found in the northern and western parts of Norway. But most of the basic terminology of the model is fully compatible with most varieties of Norwegian.

Figure 6 Pitch analysis and annotation of an utterance, according to the Trondheim model. The analysed utterance is Thorstein snakker bestandig om intonasjon (Eng. “Thorstein always talks about intonation”). The illustration is from Nilsen (1992: 31). The annotation is reproduced by the author



define the left edge of the *Foot*, as we will see in 4.1.3 below.

3.2 Word

The second level is called the prosodic word. This is probably not a relevant level if one wants to analyse different variants of spoken Norwegian, as pointed out by Abrahamsen (1998). The important exception is the assignment of tonal accent, which is most often determined by the lexical accent of the first word in each *Foot*.

3.3 Foot

The *Foot* or *Accent Phrase*⁷⁾ is the main element in the third level. With reference to the numerous studies of the Norwegian tonal accent that have solely analysed production of single words, this is the level that unfortunately often is misunderstood to be directly related to word units. But as we will see in the following example, an AP can be comprised by several words (Figure 7).

As mentioned in 4.1.1, all *Feet* (henceforth called *APs*) begin with a stressed syllable. The tonal accent is assigned to the AP (which may consist of more than one word). As we saw in Figures 1 and 2, Norwegian has two phonetically different tonal accents, 1 (t1) and 2 (t2). The tonal accent assigned to an AP is most often decided by the lexical tone attached to the first word in the AP. An example of this is the AP “hammer til å” in Figure 7, where “hammer” has the lexical tone 2 when pronounced in isolation, and therefore the AP beginning with “hammer” is fitted with tonal accent 2. But sometimes we get a shift in lexical tone, as in another AP in the utterance in Figure 7; “få på plass”, which gets t2 even though “få” is a monosyllabic word, and therefore originally has t1.

But all tonal units (TU)⁸⁾ on this level are not APs. The first two words in Figure 7 may be

pronounced without stress, and will in that case be an example of an *anacrusis*. An anacrusis is an unstressed element located at the beginning of an utterance. The three TUs following this anacrusis are APs. As we see, the three APs consist of three, three, and one word, respectively.

3.4 Intonational Phrase (IP)

The Trondheim model deals with two kinds of APs, based on degree of prosodic prominence. The most prominent AP is a *focal* AP, while the less prominent AP is *non-focal*. As we see in Figure 6, the Trondheim model signals the degree of focality with upper and lower case letters, respectively. The phonetic difference between a focal and a non-focal AP is primarily located to the right edge of the unit in East Norwegian, where a focal AP typically has a sharp rise that is not found in a non-focal AP. This is illustrated in Figures 1 and 2, where both the words uttered constitute focal APs. Studies within the Trondheim model have shown that the focal vs. non-focal dichotomy is quite crucial to the mechanisms of East Norwegian intonation, and the model has therefore included the IP level, where a focal AP normally constitutes the right edge of an IP.

3.5 Intonational Utterance (IU)

The top level of the Trondheim model, the IU, may be compared to the sentence level in writing, although this is not always true: A sentence may be pronounced as two IUs, and vice versa. One can briefly say that an IU is an entity that can be uttered on its own.

4 An Ongoing Phonetic Study of Norwegian Intonation

The study presented below is a part of the author’s doctoral project. One aim of this phonetic investigation is to describe what are the typical contours for different types of tonal units. Another aim is to represent these typical contours in a format that makes them applicable for prosodic modelling purposes within a speech synthesis system.

Figure 7 Illustration of division into tonal units based on one way of saying the declarative utterance Jeg brukte hammer til å få på plass spikeren (Eng. ‘I used a hammer (in order) to put the nail in its right place’). Borders between the tonal units are symbolised with vertical bars

Jeg brukte | hammer til å | få på plass | spikeren

⁷⁾ The *Foot* is often called *Accent Phrase* (AP) (e.g. Kristoffersen 2000, pp. 275), and has also been known as *Tonelagsgruppe* (Alnæs 1916: 92ff).

⁸⁾ TU is a collective term for *Accent Phrases* (also called “*Foot*” or “*Tonelagsgruppe*”) and *Anacruses*.

As a consequence of these aims, we have to define reasonable ways of grouping the current speech material into different types of tonal units. All the features from the Trondheim model mentioned in 4.1.1 through 4.1.5 are relevant here, in addition to other dimensions like speech style, gender, dialect, age, etc. The next sections will present some examples of this from the author's ongoing study. Some central factors, such as selection of speech data, segmentation and acoustic analysis will not be dealt with in detail in this article.

4.1 Annotation and Classification of Continuous Speech

A phonetic intonation study requires recorded speech data. This project uses recordings of continuous read speech from the Telenor speech data base *TABU.0* (Amdal and Ljøen 1995).⁹⁾ As mentioned earlier, the speech data has to be annotated within a relevant linguistic framework. In the present project, an annotation that is partly based on the description tool within the Trondheim model is arranged in a spreadsheet, where each tonal unit gets its own row, and the features (like e.g. tonal accent, focality, and number of syllables) are organized in columns. It is thus possible to group the tonal units on the basis of each of the annotated features (or combinations of these), i.e. one can sort out e.g. all APs with t2 that are focal, disyllabic and utterance-initial.

4.2 Analysis by Synthesis: Stylisation of Pitch Contours

Speech synthesis has been used by phoneticians in order to test out which are the (perceptually) important movements in intonation contours. The current project uses stylisation through synthesis of pitch contours in order to answer that question. This approach combines acoustic and auditive efforts. Studies like e.g. Werner (2000) have documented that intonation contours can be stylised quite extensively without reducing the perceptual naturalness of the melody in the speech signal. Stylisations remove artefacts like micro-prosodic "bumps" in the contour, and if the stylisations are consequently created on an auditory basis, they can be generalised into contours that can be claimed to be perceptually representative.

4.3 Stylisation Strategy

There are a lot of ways to define criteria for how the stylisation process should be carried out. We will not go into details here, but just point out some possible solutions to these questions.

A stylisation strategy can be either auditory- or acoustic based, or it can be a combination of these. Acoustic based strategies typically use algorithms in order to pin out the parts of the pitch contour where the biggest changes occur, and consequently make stylisations based on coordinates located at these points (e.g. Frid 2001). The current project uses stylisation strategies that are mainly auditory-based, as outlined below.

When stylising a tonal unit, we have to decide how many coordinates the stylisation should be based on. Should it be a fixed number of coordinates, or should we rather use a maximum number of coordinates? We can compare this question to e.g. the acoustic analyses by Haugen and Joos (1952), who partitioned their tonal units (that were single words) with t1 and t2 into three and six phases, respectively. In the ongoing project we use a different strategy from Haugen et al.: We use the same criterion for all types of TUs, including unstressed TUs, namely a maximum number of *six coordinates* (hereby called *turning points* or TPs) in the stylisation of TUs. This number of TPs is based on pilot testing that showed that the maximum number of needed TPs in order to make a *close copy*¹⁰⁾ of the melody of a tonal unit was *six*, irrespective of its length, tonal accent and complexity. Another criterion says that one shall never use more TPs than necessary, i.e. if you do not get a melody that is noticeably closer to the original by adding a TP, then the added TP is not relevant and will not be used. As a consequence, the stylised pitch contour of a TU can be based on everything from two to six TPs.

In order to perform consistent stylisations one has to have many more criteria than those already mentioned; we have to have criteria for how to define the pitch value in areas with creaky voice, etc. We will not go into this or other questions here, but just state the fact that criteria for all such instances have to be established.

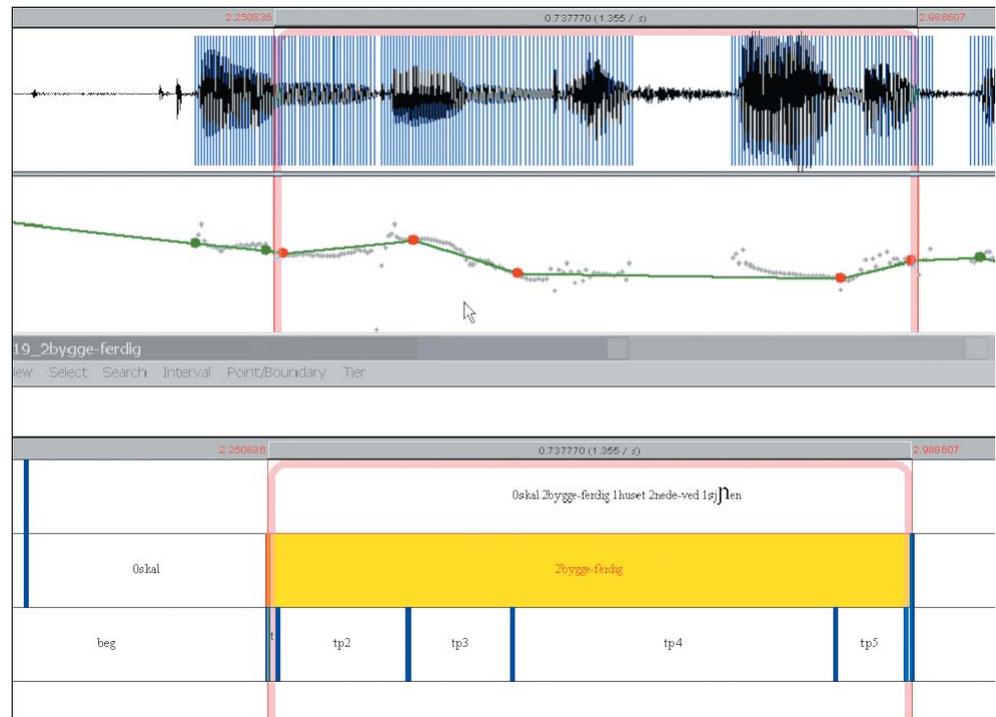
Instead we will illustrate what a stylised pitch contour may be like with the strategy that is partly described here.

Figure 8 shows a stylisation of the focal AP (²bygge-ferdig), which has tonal accent 2 and consists of four syllables. These are just some of the features that can be assigned to a TU. The main point here is that these feature data can be used in order to find out which types of TU

⁹⁾ The size and quality of the speech data is of course crucial, but will not be dealt with in this paper.

¹⁰⁾ A close copy is a speech stimulus that is modified according to a set of criteria, but still with the intention of sounding as close to the original as possible.

Figure 8 Example of a stylised version of an AP, (2bygge-ferdig), uttered by an informant from Trøndelag. From top to bottom the illustration consists of the wave form, a stylised intonation contour, and an annotated segmentation with three levels. A pitch analysis can be seen as a grey curve in the same widow as the stylisation, where the TPs can be seen as red dots

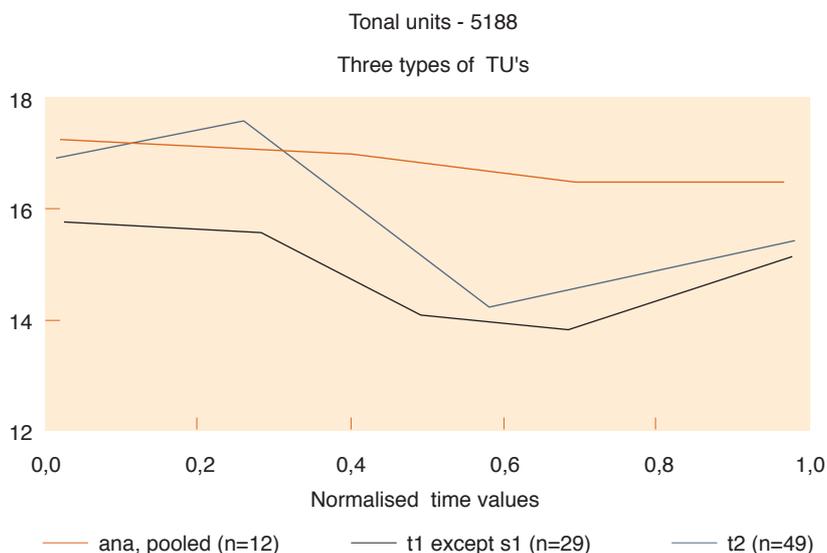


groupings that seem reasonable, based on statistical treatments of different types of TUs. Examples of some preliminary results based on such groupings are presented in the next section.

4.4 Some Preliminary Results

The possibilities for grouping TUs based on linguistic features are numerous. Three examples of such groupings will be illustrated below. The results presented here are based on analyses of speech material consisting of 19 sentences read from a manuscript by a young female from the Trøndelag area.

Figure 9 Averaged pitch contours for anacruses (ana) and APs with tonal accent 1 (t1) and tonal accent 2 (t2), respectively. Normalised time scale (x-axis), and pitch scale in semitones relative to 100 Hz



First some basic numbers: The 19 read sentences consist of 111 TUs; 12 are anacruses (ana) and the remaining 99 are APs. 50 of the APs have tonal accent 1 (t1), and 49 APs have tonal accent 2 (t2). The averaged contours for these three groups are illustrated in Figure 5.¹¹⁾

The contours in Figure 9 show that these three groups seem to generate different pitch contours. The anacruses seem to have a higher overall pitch than the two AP groups, which is reasonable since they are utterance-initial, and that there is a declination (a gradual lowering of the pitch) throughout the utterances. We also see that the APs generate more tonal movement than the anacruses. The main difference between these two AP groups (i.e. t1 and t2) is located to the left edge, where t2 seems to have an initial rise, in contrast to the falling start of t1.

Figure 10 shows the averaged tonal realisations of APs with tonal accent 1, grouped with reference to syllable number. The contours indicate a steeper fall at the left edge for the polysyllabic APs than for the monosyllabic ones. We might remember that Vanvik (1956) claimed that monosyllabic APs (i.e. the *t1s1* category in Figure 10) generate a third phoneme, i.e. that their tonal realisation should not be referred to as tonal accent 1. This might be a typical *trønder* or Mid-Norwegian phenomenon, since similar studies of South East Norwegian have documented no such difference when comparing

¹¹⁾ Monosyllabic APs are excluded from the t1 group, since a t2 AP cannot be monosyllabic.

monosyllabic and polysyllabic APs with tonal accent 1.

Figure 11 shows the TUs with tonal accent 1, grouped with reference to the degree of focality. These contours confirm what is postulated in the Trondheim model: The phonetic difference between focal and non-focal APs in low tone dialects is located to the right edge, where a focal AP typically has a sharp rising contour, compared to the relatively flat contour of a non-focal AP.

As Figures 9–11 illustrate, the tonal contours are shaped quite like the ones shown in earlier studies. But the measuring unit of the time scale in these figures has not been used in tonal studies of Norwegian before, except in the studies by Fintoft et al. A normalisation of the absolute time values from the analysis makes it possible to compare TUs with different durations. This legitimates comparisons like the one illustrated in Figure 10, where the length of the APs, measured in number of syllables, is the only feature separating the four groups.

There are of course potential weaknesses associated with the use of normalised time scales. One possible weakness is that this scale does not have any syllabic information, which could be desirable since there is no doubt that the syllable is a crucial entity in our context.

In this study, the measure unit used for pitch is *semitones*, which is logarithmic and therefore more similar to the way we perceive different tone heights etc., if we compare it to the Hertz scale, which is more frequently used in works of this type.

4.5 Further Work

Analyses such as the one described above are being carried out for speakers from different dialect regions. The speakers have read the same manuscript. This will make it possible to suggest what are typical tonal properties for these regions, in addition to making comparative analyses between the regions in order to find out what makes them tonally different.

Furthermore, the production analyses will form the basis for listening tests where listeners from the different dialect regions will tell to which degree the different production results are perceptually salient. We will also try to find out what are the perceptually most important intonation differences between the dialect regions.

4.6 Conclusion

When we compare the study in the last section to the earlier studies by e.g. Vanvik, and by Fintoft et al., we see a number of methodological differ-

Tonal units - 5188
t1 units grouped according to syllable number

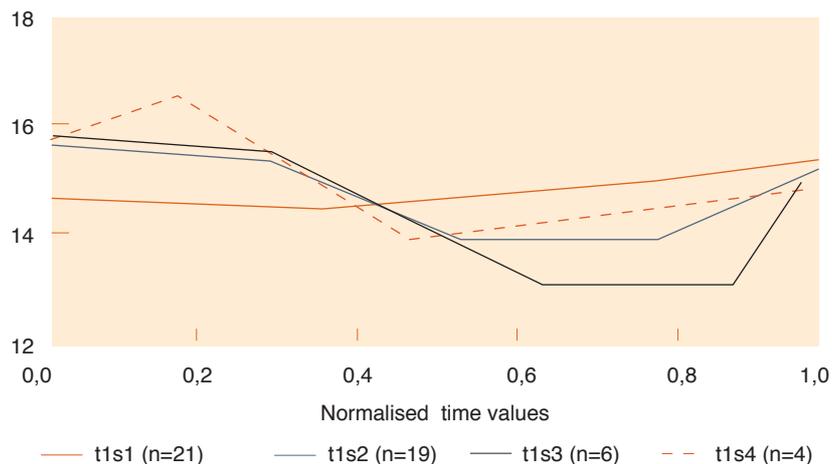


Figure 10 APs with tonal accent 1 (t1), grouped with reference to syllable number. s1 = one syllable, s2 = two syllables, etc.

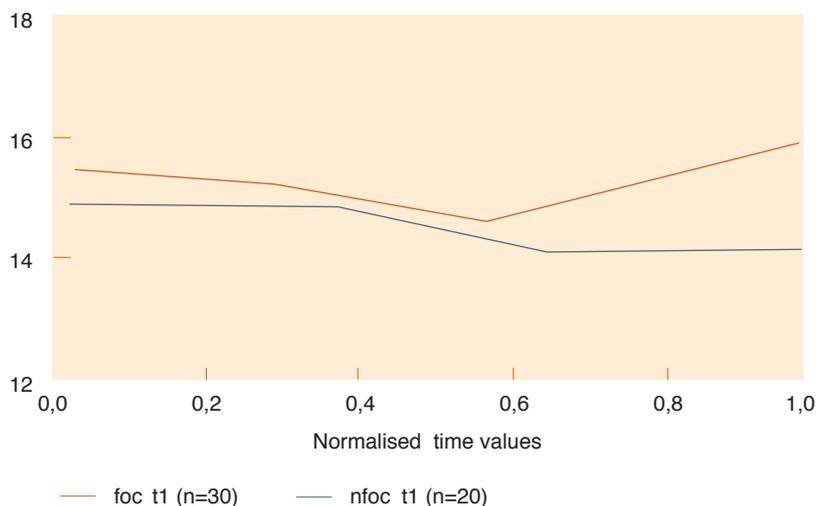
ences, as well as their different ways of presenting their results. With respect to speech data, both Vanvik and Fintoft et al. carried out their analyses exclusively from single word units, and with a very limited variation in the number of syllables those words consisted of. If a study focuses on analyses of disyllabic words, the results should not automatically be considered to be relevant for words consisting of any other number of syllables.

Another methodological point is the choice of analysis tools. Vanvik drew graphs with a pencil based on what he heard, while Fintoft et al. trusted their pitch meter, paying practically no attention to the perceptual dimension in their studies.

As Malmberg pointed out, the most complete phonetic studies combine linguistic and impressionistic skills with acoustic/instrumental mea-

Figure 11 Focal (foc) and non-focal (nfoc) APs with tonal accent 1

Tonal units - 5188
Tonal accent 1, focal and non-focal



suring. The study of the author that is briefly presented in this article can be said to hold this combination: The production analysis is based on entities defined with a linguistic annotation and classification system (i.e. the Trondheim model); each entity is analysed through an auditory-based tonal stylisation of the original recording. The data from the coordinates describing the stylisations are taken from acoustic analyses. As mentioned initially in this article, the aim behind the development of this methodology is two-fold: firstly, we want to do a proper phonetic investigation of Norwegian intonation, and secondly, we wish this study to be of some interest within the context of intonation modelling in speech synthesis applications.

5 Literature

Abrahamsen, J E. Remarks on the oxytonic accentual pattern in a West Norwegian dialect. In: Werner, S (ed.). *Nordic Prosody VII*, 23–33. Frankfurt am Main, Peter Lang Europäischer Verlag der Wissenschaften, 1998.

Almberg, J, Husby, O. The relevance of some acoustic parameters for the perception of a foreign accent. In: James, Allen and Jonathan Leather (eds.). *New sounds 2000 – Proceedings of the Fourth International Symposium on the Acquisition of Second-Language Speech*. Amsterdam, University of Klagenfurt, 2002.

Alnæs, I. *Norsk sætningmelodi : dens forhold til ordmelodien : en undersøkelse av østnorsk riksmåal*. Kristiania, Aschehoug, 1916.

Amdal, I, Ljøen, H. *TABU.0 – en norsk telefontaledatabase*. Kjeller, Norway, Telenor R&D, 1995. (FoU R 40/95)

Cruttenden, A. *Intonation*, second edition. Cambridge University Press, 1997.

van Dommelen, W, Fretheim, T, Nilsen, R A. The Perception of Boundary Tone in East Norwegian. In: Werner, S (ed.). *Nordic Prosody VII*. Frankfurt am Main, Peter Lang Europäischer Verlag der Wissenschaften, 73–86, 1997.

Fintoft, K, Mjaavatn, P E. Tonelagskurver som målmerke. *Maal og Minne*, 66–87, 1980.

Foldvik, A K. Realisasjonen av r i norsk. In: Jahr, E H and Lorents, O (eds). *Fonologi/Phonology*. Oslo, Novus, 1981, 319–327, 1978.

Fretheim, T, Nilsen, R A. Romsdal Intonation : Where East and West Norwegian Pitch Contours meet. In: Niemi, J (ed.). *Papers from the eleventh Scandinavian conference of linguistics*, Joensuu, 2, 442–458, 1989.

Fretheim, T, Nilsen, R A. Alternativspørsmål : opp som en løve, ned som en skinnfell. *Norsk Lingvistisk Tidsskrift*, 1/2, 89–104, 1988.

Frid, J. Prediction of intonation patterns of accented words in a corpus of read Swedish news. In: Karlson, A and van de Weijer, J (eds.). *Papers from Fonetik 2001 held at Örenäs, May 30 – June 1, 2001*. Working Papers 49, 42–45, 2001. Lund, UB Media.

Haugen, E, Joos, M. Tone and intonation in East Norwegian. *Acta Philologica Scandinavia*, 22, 41–64, 1952.

Husby, O, Almberg, J. Russeres og nordmenns persepsjon av en russisktalendes norske andrespråk. In: Moen, I et al. (eds.). *MONS 9, Utvalgte artikler fra Det niende møtet om norsk språk i Oslo 2001*. Oslo, Novus Forlag, 2002.

Jahr, E H, Lorentz, O (eds). *Prosodi/Prosody*. Oslo, Novus, 1983, 63–67, 1925.

Jahr, E H, Lorentz, O (eds). *Fonologi/Phonology*. Oslo, Novus, 1981, 319–327, 1978.

Jakobson, R, Halle, M. *Fundamentals of Language*. 's-Gravenhage, Mouton, 1956. (Janua linguarum, Series minor, 1.)

Knudsen, K. *Haandbog i dansk-norsk sprog-lære*. Kristiania, Abelsted, 1856.

Kristoffersen, G. *The Phonology of Norwegian*. Oxford, Clarendon Press, 2000.

Malmberg, B. Levels of Abstraction in Phonetic and Phonemic Analysis. *Phonetica*, 8, 220–243, 1962.

Mo, E. Tonelagsvald og stavingar. In: Jahr, E H, Lorentz, O (eds). *Prosodi/Prosody*. Oslo, Novus, 1983, 63–67, 1925.

Nilsen, R A. *Intonasjon i interaksjon : sentrale spørsmål i norsk intonologi*. Trondheim, Universitetet i Trondheim, 1992. Doctoral dissertation.

Selmer, E W. Tonelagsproblemer. *Maal og Minne*, 1, 180–188, 1954.

Selmer, E W. Noen bemerkninger om den musikalske aksent i dens forhold til den sterke skårne og svakte skårne aksent. In: Jahr, E H, Lorentz, O (eds). *Prosodi/Prosody*. Oslo, Novus, 1983, 68–77, 1925.

Storm, J. *Illustreret Nyhedsblad*, 40, 42, 1860.

Storm, J. Om tonefaldet (Tonelaget) i de skandinaviske Sprog. In: Jahr, E H, Lorentz, O (eds).

Prosodi/Prosody. Oslo, Novus, 1983, 30–39, 1874.

Vanvik, A. Om fonetiske villfarelser. *Småskrifter fra Fonetisk institutt*, Universitetet i Oslo, 1983. (In series)

Vanvik, A. A Phonetic-Phonemic Analysis of standard Eastern Norwegian, Part II. *Norwegian Journal of Linguistics*, 27, 119–164, 1973.

Vanvik, A. A Phonetic-Phonemic Analysis of standard Eastern Norwegian, Part I. *Norwegian Journal of Linguistics*, 26, 101–139, 1972.

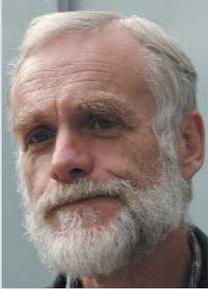
Vanvik, A. *A Phonetic-Phonemic Analysis of the Dialect of Trondheim*. Oslo, Universitetsforlaget, 1966.

Vanvik, A. Norske tonelag. In: Jahr, E H, Lorentz, O (eds). *Prosodi/Prosody*. Oslo, Novus, 1983, 209–219, 1956.

Werner, S. *Modelle deutscher Intonation: zu Vergleichbarkeit und empirischer Relevanz von Intonationsbeschreibungen*. Joensuu, Joensuu yliopisto, 2000. (Doctoral dissertation)

Prosodic Unit Selection for Text-to-Speech Synthesis

JON EMIL NATVIG AND PER OLAV HEGGTVEIT



Jon Emil Natvig (57) is senior research scientist at Telenor R&D. He is currently working in the Future Wireless World research group with special interests in speech technology and voice interfaces for future mobile systems. In recent years Jon Emil Natvig has been involved in developing the research into Text-to-Speech system for Norwegian – Talsmann[®]. His current research interests are centered around techniques for natural sounding speech generation for speech synthesis. Natvig graduated from the Norwegian Institute of Technology (NTH) as Siv.Ing. (MSc) in 1970 and Dr.Ing. (PhD) in 1975.

jon-emil.natvig@telenor.com



Per Olav Heggveit (37) is a research scientist at Telenor R&D. He graduated from the Norwegian University of Science and Technology as Siv.Ing. (MSc) in 1989. In 1990–1991 he worked as product manager for speech products at Falck Produkter AS. While finishing part-time studies in computational linguistics at the University of Oslo, he started in 1991 as a research scientist at Telenor R&D working mainly with text-to-speech synthesis, and in particular with text analysis and prosody modelling. Heggveit's recent work includes IP-based voice services, unit selection synthesis and voice enabled web services.

per-olav.heggveit@telenor.com

We present work done on a novel approach for prosody generation in text-to-speech synthesis. The approach is based on selection of appropriate intonation units in a corpus of units extracted from natural speech. The units are defined on the basis of a phonological model of Norwegian intonation. The theoretical background, the model itself and a demonstrator are described. The new model is compared to an existing prosody model and prosody copied from natural speech in a subjective listening test. The result shows that the new model is clearly preferred compared to the existing prosody model, but still not as good as prosody copied from natural speech.

1 Introduction

Prosody¹⁾, and in particular intonation, plays a key role in the perceived naturalness of synthetic speech [1, 2]. In TTS systems, generation of prosody usually takes place in two separate steps: (i) first an abstract description of the sentence prosody is derived on a linguistic level, typically represented as the prominence level of each word, which word that has sentence accent (focus), phrase structures and boundaries. (ii) Given this information, the next step calculates the acoustic parameters (F0 contour, phoneme durations, pauses, etc.), which are the acoustic correlates of the linguistic factors to produce the acoustic realization of the intended prosody.

The following two procedures may be considered to represent the extremes in prosodic realisation procedures in current TTS systems:

- 1 *Rule based* methods with unit modification
 - a Compute quantitative target values for the acoustic parameters based on the linguistic prosodic tags: As a minimum, this includes phoneme durations and F0 values. Other acoustic correlates of prosody are amplitude and vowel quality.
 - b Search the speech corpus and retrieve segmental units. Typically, the units are diphones or phonemes recorded in a fixed phonemic and prosodic context.
 - c Modify the segmental units to attain the specified target values computed in step a.
- 2 *Corpus based* method
 - a Search and retrieve units with matching prosodic tags from a suitably tagged corpus.
 - b Concatenate units with no modification

Many TTS systems use procedures which are a combination of these approaches, e.g. a corpus

based method could include some unit modification in cases where no satisfactory units could be found in the corpus. Rule based methods could use context dependent units to reduce the need for modification.

Corpus based unit selection synthesis represents the current state of the art in TTS. In these methods, prosody synthesis comes as a by-product of the unit selection process. Corpus based synthesis methods have been shown to produce mostly natural-sounding prosody. However, these methods are prone to occasionally produce unnatural prosody due to loopholes in the unit coverage of the database. Coverage is one of the main challenges of corpus based synthesis methods, this is because many speech and language features exhibit the so-called *Large Number of Rare Events* (LNRE) distribution property [16, 17]. This means that while some units are quite frequent, a vast majority of the possible units are extremely rare. Even though, the probability for a *particular* rare unit to occur in an arbitrary input sequence is extremely low, the possibility that *some* rare unit will occur is near certainty.

Uneven performance is one characteristic of state of the art unit selection based speech synthesis [17]. When a good unit sequence is available in the database, the speech output quality is close to natural speech. But very often, stretches of almost perfect speech are interrupted by some very poor unit, degrading the perceived voice quality.

Increasing the size of databases will lead to better synthesis. With more data it becomes more likely that a unit closer to the target may be found. But one important limiting factor in the construction of speech databases will always be the difficulty in having a speaker deliver the necessary speech material in a consistent manner over many hours.

¹⁾ Prosody refers to suprasegmental features such as pitch (fundamental frequency variations or intonation), loudness and rhythm.

Rule based prosody systems, on the other hand, achieve less natural prosody, but perform more consistently. The main drawbacks of the rule based approach are that rule development is a time-consuming and difficult task, that tuning the rules to a specific speaking style is rather intricate, and that the set of rules may be large and complex. The prosody generated by rule based methods may be less varied than natural speech giving a repetitive speaking style.

The research challenge in rule based prosody generation is to predict detailed pitch contours on the segmental level. Typical rule based approaches rely on linear segments between pitch target points, without the natural fluctuations seen in natural speech.

The TTS system developed by Telenor R&D, is a “conventional” TTS system using diphone concatenation for speech generation [11]. The current linguistic-prosodic module of Talsmann is rule based and processes the input text sentence by sentence. The high level linguistic-prosodic analysis is based on the phonological model for Norwegian described in section 2, using tonal accent units as the basic unit. The rules have been derived from an analysis of a limited set of simple sentences [12].

The main weakness of the rule based method used in Talsmann is that it is not able to render the natural variation observed in natural speech and leads to a rather monotonous and stereotyped intonation.

The approach reported here was initiated from informal tests transplanting intonation contours from natural speech on synthetic speech. This resulted in a significantly better synthetic speech (comparable to “GSM speech”). This suggested that a (prosodic) unit selection model, transplanting F0 (fundamental frequency) contours from natural speech could be a promising approach to obtaining a better TTS quality in our system.

In this paper we report a “proof of concept” study to investigate the feasibility of this prosody transplantation approach [10]. We propose to use the same *prosodic accent units* as used in the rule-based model, for the unit selection approach. The approach consists of a simple concatenation of stored f0 contours similar to [3, 4, 5], thus avoiding the computation of the detailed acoustic specification at the segmental synthesis unit level.

In section 2 we give a presentation of the phonological model for Norwegian intonation which is the basis for our model, and section 3 describes the method in more detail: Candidate F0 contours are stored in a searchable lexicon. At synthesis time, the utterance is represented as a sequence of accent units. For each accent unit, a context dependent search is carried out and the candidate unit that is the best match with respect to the properties of the target accent units of the utterance is selected. The pitch contour over the utterance is determined by linear interpolation between a set of target pitch values located around the accented syllables. The quality of the generated intonation was evaluated in a subjective test, and compared to the original rule based approach as well as “natural” intonation copied from natural speech. This is reported in section 4.

2 A Phonological Model of Norwegian Intonation

Our work is based on the “Trondheim Model”, a phonologic model for representation of Norwegian intonation [6, 7]. According to this model, Norwegian intonation can be described in the following hierarchic model:

The basic unit of the model is the *Accent Unit* (AU). The Trondheim model uses the term Tonal Foot or Foot. An Accent Unit consists of an accented syllable (typically carrying the *Tonal Accent*, the characteristic pitch movement, of a stressed word) followed by a sequence of unaccented syllables. An AU is terminated by a following AU (i.e. an accented syllable) or a phrase boundary.

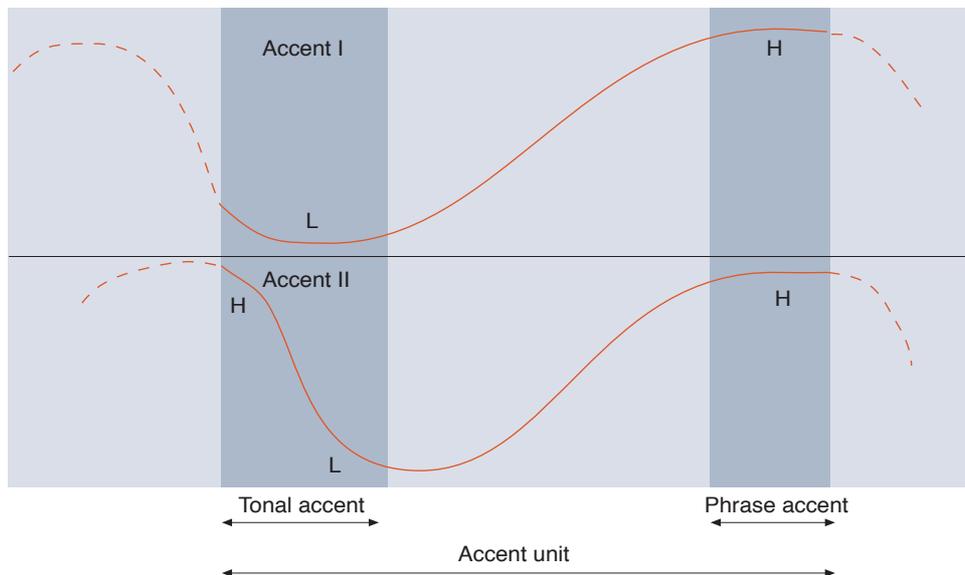
The AU also carries a *Phrasal Accent*, which in East Norwegian is manifested as a tonal rise (LH)² towards the AU-final syllable. The level of the final H tone signals the degree of prominence. The most prominent unit, the focal AU, is supposed to have the highest pitch value.

Any phrase initial unaccented syllables are classified as “AU external”. AU external syllables may also occur inside an utterance, e.g. after a boundary.

A characteristic feature of the intonation in many Norwegian dialects is the distinction between two tonal accents or tonemes. The tonal accent is a distinctive feature in Norwegian. Words with the same phonetic content may have different meaning depending on the accent, e.g. the word *rosen* means “the praise” and “the rose” when uttered with accent 1 and 2 respectively. The

²) We use the TOBI notation (<http://www.ling.ohio-state.edu/~tobil/>) to indicate tonal level and movements. The basic tone levels are high (H) in the local pitch range versus low (L) in the local pitch range. L % or H % indicate a final boundary low and high tone, which occurs at every full intonation phrase boundary.

Figure 1 Characteristic pitch contours for accent I and II in East Norwegian



tonal accents in East Norwegian are typically realised as shown stylised in Figure 1: Accent I is characterised by a low level (L) in the first syllable of the accent unit, whereas accent II is a HL transition through the first syllable. In a sentence context, accent I can also be seen as a H-L transition from the H level of the phrase accent at the end of the previous AU. As shown in Figure 1, the two accents can be regarded as time-shifted versions of the same contour.

In the Trondheim Model accent units are grouped into *Intonation Phrases* (IPs), which is the next level in the intonation hierarchy. Each IP consists of a sequence of non-focal AUs and is terminated by a single focal AU, i.e. the AU having the highest Phrasal Accent.

The IP is the domain for *focal accentuation* and for *declination*, i.e. a gradual lowering of the baseline for both H and L tones in both tonal and phrasal accents. In East-Norwegian, which is the dialect covered by our model, this declination typically takes place in AUs following the last focal foot of the IP as illustrated in Figure 2.

The highest phonological level of the model is the *Intonation Utterance* (IU), consisting of one or two IPs. (The number two is dictated by prag-

matic considerations rather than by the intonation model). In read speech, the IU normally corresponds to a sentence. The IU is the domain of the utterance final boundary tone: H % for interrogative and L % for declarative utterances.

Standard notation in the Trondheim model is to enclose intonation units in brackets with their type (F, IP, IU) given as a subscript by the closing bracket. A number in superscript by the opening bracket of AUs indicates the tonal accent type (1 or 2).

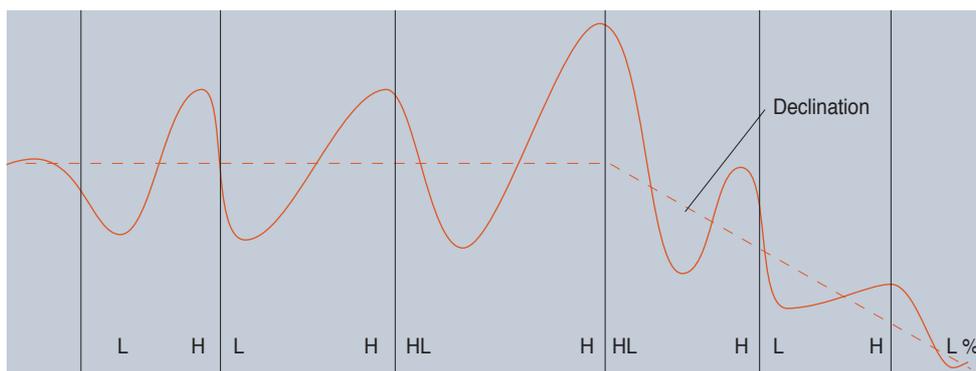
The examples below illustrate this notation used on the utterance “*I en ryggsekk hadde han kjøtt for tusen kroner*” (In a backpack he had meat for 1000 kroner).

Example:

((i en) (²RYGGsekk_F)_{IP}) (²hadde han_F)
 ((¹kjøtt for_F) (¹TUSEN_F) (²kroner_F)_{IU})_{IP} L %

Here the accent units are grouped into two intonation phrases, except for the two initial unaccented syllables forming an external group. According to our model, declination will normally take place after the second focal AU.

Figure 2 Characteristic pitch contours for Norwegian intonation in sentence context



The tonal hierarchy can also be represented as a tree structure as is shown in Figure 2. In the Figure the symbol σ indicates syllables and we also include a tone level layer (L/H) to illustrate the tonal movements associated with the accents in the case of East-Norwegian.

3 System Description

3.1 Speech Database

Since we were mainly interested in a proof of concept, we chose to use an existing speech database for this work. The speech database PROSDATA [8] used for these experiments consists of 502 Norwegian sentences read aloud by a female speaker, the duration is approximately 30 minutes. The recording was made in a studio, using a sampling rate of 16 kHz. The data have been manually segmented in terms of phonemes, syllables and words. The sentences are read in an informative news style, with a pleasant but neutral voice.

The database includes word prominence ratings and break indices which have been determined by subjective evaluation. Figures 4 and 5 show the coding of these parameters.

An automatic labelling of Accent Units based on the prosodic model in section 2 was carried out as follows: For each word with prominence 3 or 4, we locate the accented syllable of the corresponding word, using a pronunciation lexicon. A foot is then constructed starting with this syllable and including all following unaccented syllables and ending at the syllable before the next accented syllable or at the end of a word with break index greater than 1. Syllables not included in a foot in this procedure are grouped into "AU external" groups.

For each sentence a smoothed f0 curve has been determined which makes it possible to automatically determine the necessary pitch values in the intonation lexicon.

3.2 Intonation Lexicon (Defining and Collecting Units)

A searchable lexicon of tonal feet was automatically extracted from the PROSDATA corpus.

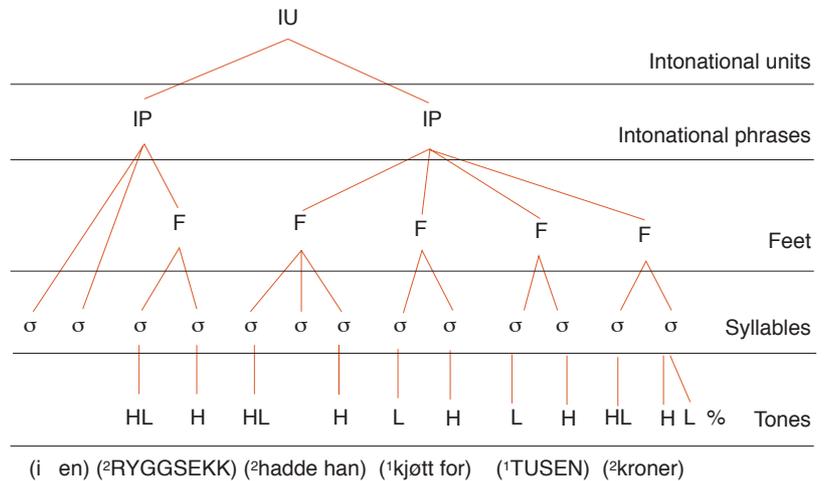


Figure 3 Phonological tree structure

For each accent unit we stored

Context parameters:

Properties of previous unit (Note 1)

- Unit type: normal AU, focal AU, external unit
- Position information: initial, medial, postfocal
- Average pitch of last syllable of previous unit

Properties of the current unit

- Unit type, normal AU, focal AU, external unit
- Position information: initial, medial, postfocal, final
- Toneme: 1 or 2 or none (if external group)
- The number of syllables in the unit
- Syllable structure of accented syllable (V, CV, VC ...)
- The phoneme string for each syllable in the unit
- The pitch value in Hz at syllable start and end, and vowel start and end for the first, second and last syllable of the accent unit (Note 2)
- Average pitch of last syllable

Properties of the following unit (Note 1)

- Unit type, accent unit (AU) or external unit
- Position information: medial, postfocal, final
- Toneme : 1 or 2 or none (if external group)
-

Notes:

- 1 When concatenating pitch contours from accent units we want to ensure F0 continuity at the boundaries. In this process we need to

- 1 No prominence
- 2 Moderate prominence
- 3 Distinct prominence (accented)
- 4 Focal prominence

Figure 4 Word prominence scale

- 1 No break
- 2 Moderate sentence internal break
- 3 Distinct break
- 4 Utterance break

Figure 5 Word break index scale

Table 1 Factors and factor levels in unit selection search parameters

| | Preceding unit | Current unit | Following unit |
|-----------------------------|--|--|---|
| Search parameters | Unit type {f,F,e} Position code {i,m,f,p} | Unit type {f,F,e} Position code {i,m,f,p} Accent {1,2} | Unit type {f,F,e} Position code {i,m,f,p} Accent {1,2} |
| Selection parameters | Average pitch in last syllable | Phoneme string Number of syllables in accent unit Syllable structure of accented syllable Pitch values for first, second and last syllable: At syllable start and end At vowel start and end Average pitch in last syllable | |

consider several contextual effects. At the left boundary we need to consider the pitch level in the last syllable which is mainly determined by the phrasal accent level (“normal” or focal) and also whether the unit is in the declination part of the utterance (i.e. postfocal) or not. For the right hand boundary, we only need to take into account the tonal accent type and the utterance position (postfocal or not). The assumption being that the phrasal accent of the following AU will not influence the boundary behaviour of the current unit significantly.

- 2 According to the intonation model for East-Norwegian, the tonal accent is associated with the first syllable of the AU, and the phrase accent with the last syllable. We therefore argue that the essential part of the intonation contour can be described by the three consecutive syllables: the last syllable in the preceding foot and first and second syllables of the foot in question. Any intermediate syllable within a foot is regarded as less important and can be modelled by interpolation.

Table 1 shows the parameters available stored in the lexicon.

3.3 Prosody Synthesis

3.3.1 Prosodic Analysis (Specifying Target Units)

The first step in the prosody generation process is to specify the intended prosodic phrasing and focusing of the input text. In a TTS system this is typically done by text analysis resulting in some kind of symbolic prosodic representation. In this experiment, this analysis was done manually.

We use a symbolic prosodic representation of each accent unit in the following format:

Initial unit:

`_<unit type>[<phoneme string>]`

Intermediate units:

`<unit type>[<phoneme string>]`

Final units:

`<unit type>[<phoneme string>]%`

`<unit type>` is one of the following:

- `f1, f2`: Normal AU (accent I or II)
- `pf1, pf2`: Postfocal AU (accent I or II)
- `F1, F2`: Focal AU (accent I or II)
- `e`: Unit consisting of external syllables

`<phoneme string>` is a string of Norwegian SAMPA phonetic.

Example:

The sentence “Men i DETTE tilfellet så han trolig ikke bilen”. (In THIS case he probably did not see the car.) results in the symbolic representation of five units described in Figure 6.

```

_e[men$i]
F2[det$@$%til$fel$@]
pf1[sO:$hAn]
pf2[tru:$li$%ik$@]
pf1[bi:l$@n]%

```

Figure 6 Symbolic representation of the sentence “Men i DETTE tilfellet så han trolig ikke bilen”

3.3.2 Lexicon Search (Finding Units)

Given the list of target prosodic units the list is traversed from left to right. The key used to retrieve candidate units from the lexicon is built as follows:

$\langle left_unit, cur_unit, right_unit \rangle$

where:

$left_unit$ is unit type of preceding unit
 {e,f,F,pf} leaving out the accent information;

cur_unit is unit type;

$right_unit$ is unit type.

The example in the previous section would result in the search codes as shown in Table 2.

Normally, the lexicon search will result in a list of candidate units satisfying the search criteria. If no candidate units are found, the criteria in the unit selection key are gradually relaxed until at least one candidate is found.

3.3.3 Selecting Units

For this experiment a simplified selection strategy was implemented: the list of units is traversed from left to right. In each position, the list of candidates found in the search are scored according to the appropriateness of the units in the local context alone.

The following factors were taken into account in the selection process:

- Syllable structure of the first (accented) syllable. The syllable structure of the accented syllable is scored according to the similarity to the target unit. The scoring strategy was based on a simple heuristic approach: Exact phonemic match will give the highest score, a syllable structure match a somewhat lower score, etc.
- Number of syllables in the accent unit. Monosyllabic candidate units are excluded when the target is polysyllabic, and vice versa.
- Pitch continuity at concatenation point. In this experiment we did not implement a concatenation cost factor: If the average F0 level in the syllable at the concatenation point differs significantly (set ad-hoc to > 20 %) from the average F0 level in the last syllable of the unit selected in the previous step, the candidate is simply discarded.

An overall candidate score is calculated by weighting the syllable structure score (0.6) and the unit length score (0.4) and the unit having the highest score is retained.

| Search key | Meaning |
|-----------------|--|
| [, _e, F2]: | Initial AU external syllable group, followed by focal accent 2 AU |
| [_e, F2, pf1]: | Focal accent 2 AU, preceded by AU external group and followed by postfocal accent 1 AU |
| [F, pf1, pf2]: | Postfocal accent 1 AU, preceded by focal AU and followed by accent 2 (postfocal) AU |
| [pf, pf2, pf1]: | Postfocal accent 2 AU, preceded by postfocal AU and followed by accent 1 AU |
| [pf, pf1%,]: | Final postfocal accent 1 AU, preceded by postfocal AU |

3.3.4 F0 Curve Generation

Table 2

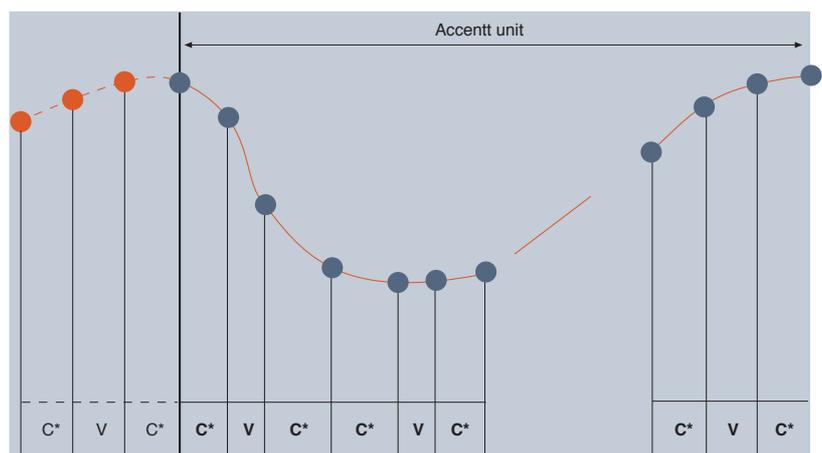
The database search results in a set of f0 target points located in the three syllables around the first syllable of each accent or external unit. The overall f0 contour was determined using linear interpolation and the pitch values were then mapped on the individual phonemes after the duration calculations. Experiments using spline interpolation did not result in any perceivable improvement compared to linear interpolation in this experiment. Figure 7 illustrates the interpolation procedure.

The speech material in the PROSDATA database is read by a female voice, whereas the TTS system has a male voice diphone library. In [15] it is suggested that pitch movements in different pitch registers are considered equivalent when equal on a psychoacoustic scale representing the frequency selectivity of the auditory system. We used the following formulas from [15] to transform to and from a psychoacoustic scale E :

$$E(f) = 16.7 \log_{10}(1 + f/165.4), \quad (1)$$

where E is the psychoacoustic scale and f is the frequency in Hz. The transformation of a particular pitch movement ($f_j - f_{j0}$) from a female pitch register to a male pitch register ($f_m - f_{m0}$) is done as follows:

$$E(f_j) - E(f_{j0}) = E(f_m) - E(f_{m0}) \quad (2) \quad \text{Figure 7 F0 curve generation}$$



$$E(f_m) = E(f_p) - E(f_{j0}) + E(f_{m0}) \quad (3)$$

$$f_m = 156.4 (10^{0.06E(f_m)} - 1) \quad (4)$$

The following base pitch values for female and male voices respectively, were used: $f_{j0} = 180$ and $f_{m0} = 85$ Hz.

3.3.5 Duration Modelling

A Classification and Regression Tree (CART) model for phone durations was trained using the software tool Wagon from The Edinburgh Speech Tools Library [11]. The database PROSDATA [8] used for intonation modelling is also used for the duration modelling [9, 10]. The model is trained on 90 % of the database and tested on the other 10 % (every tenth sentence). The model input factors were selected among the factors available in the phonological representation used as input to the intonation model.

The following factors are found important for predicting phone duration in our data and have been used to train the CART model:

- Current phoneme
- Current phoneme class
- Next phoneme class
- Previous phoneme class
- Syllable stress (unstressed, stressed, accented)
- Syllable break index (3 levels)

The position of a syllable in higher level units such as accent unit and length of the higher level units containing the syllable did not contribute significantly to the prediction of phone durations.

Linguistic level factors on word or sentence level do not improve the overall performance of the prediction. This is probably due to data sparsity when applying CART in phone duration modelling. The mean prediction error of the model is 14 ms and the corresponding correlation is 0.773.

4 Perceptual Evaluation

We carried out a subjective test comparing synthesised sentences using the new intonation model with sentences synthesised using the rule based model in our TTS system Talsmann and with sentences with intonation directly copied from natural speech.

4.1 Stimuli

4.1.1 Sentences

Ten relatively short sentences from the Prosdatabank were selected (Table 3). These sentences were then excluded from the prosody database when extracting the intonation lexicon described in section 3.2. The sentences were

| Sent. id | AU structure ³⁾ | Text |
|----------|--|--|
| 1 | F1 (2) pf1(3) pf1(4) pf2(2) | Deretter traff han personbilen i siden. |
| 2 | e(2) F2 (5) pf1(2) pf2(4) pf1(2) | Men i dette tilfellet så han trolig ikke bilen. |
| 3 | e(2) F2 (2) f2(2) f1(2) F1 (2) pf2(2) | I en ryggsekk hadde han kjøtt for 1000 kroner. |
| 4 | f2(2) F2 (3) pf1(3) pf1(2) pf1(2) | Ingen mennesker ble skadd eller omkom i brannen. |
| 5 | e(2) f1(3) f1(2) f1(3) F2 (3) z(...) pf1(2) pf1(2) pf2(3) | Isteden ble presten møtt av et flammehav da hun kom til Klokkarvik. |
| 6 | F2 (3) f2(2) f2(4) f2(6) F2 (2) | EU har ingen kvoter eller støtteordninger for geitmelk. |
| 7 | F1 (1) pf1(1) pf1(2) pf1(1) pf2(4) pf1(3) | En mann var igjen inne i forretningen. |
| 8 | F1 (2) f1(3) f2(2) F1 (1) pf2(2) | Den har vist seg å være svært nyttig. |
| 9 | e(2) f1(8) f1(2) f1(2) F2 (2) z(...) e(2) pf2(2) pf1(2) pf1(4) | Kriminalpolitisenralen er bedt om teknisk bistand for å finne fram til brannårsaken. |
| 10 | f2(2) f1(1) F2 (3) z(...) f1(3) f1(2) F2 (2) pf1(1) pf2(3) | Hjemme hos to andre menn fant politiet enda mer kjøttvarer. |

Table 3 Description of test sentences in terms of accent units and no. of syllables

³⁾ Notation. <Unit type><tonal accent>(<# syllables>); z(...) = pause

selected to represent a variation in factors like position and number of focal AU (initial, medial, final), tonal accents 1 and 2. The number of syllables in AUs varied from 1 to 8.

4.1.2 Prosodic conditions

Three different prosody models were applied to the ten sentences. This resulted in lists of phonemes, durations and pitch values, which were fed into the same concatenative speech generator taken from the Talsmann TTS [11].

The following three models were compared:

Copy synthesis from the original (CPY)

The original prosody (intonation and durations) was copied from the recording of each sentence and shifted down in pitch to match the male voice of the synthesizer.

The new prosody model (NEW)

Each of the ten sentences were manually analysed in terms of accent units and the result fed into the intonation and duration models described in section 3, to produce intonation contours and durations.

The Talsmann TTS model (TTS)

The input to the standard TTS system Talsmann was the sentence texts with emphasis tags on the focal words to force the synthesizer to deliver the same high-level prosodic information. The Talsmann TTS uses rule-based intonation modeling and a simple regression model for durations trained on a limited corpus of short declarative sentences.

4.2 Experimental Procedure

We used computer based experimental set up where the subjects could perform the test at their own leisure by accessing a web-page via the company intranet using their own PC. The test was run automatically and for each sentence, a window as shown in Figure 6 was presented. A multimedia panel and a corresponding row of buttons were presented for each intonation condition for that sentence. The subjects could play the sound files by clicking on the corresponding buttons. The identity of the models was unknown to the subjects and the position in the panel was randomized for each new subject.

The subjects were told to listen to the three samples by clicking the corresponding play button as much as they wanted. The task was to rank the three intonation examples from perceived as “Most Natural” to “Least Natural”. The subjects were forced to give rank orders in all cases and were not allowed to proceed unless a valid response had been given. Two initial sentences were presented as familiarization to the test and were not included in the analysis.

The subjects taking part in the experiment were volunteers among Telenor R&D staff. None of them were in any way involved in the project.

4.3 Results

For each condition we obtain a rank order from each subject. Each rank order can be seen as three pair comparisons. Table 4 shows the percentage of judgements where a prosody model was rated better than another as an average over all test sentences.

All preferences in Table 4 are statistically significant at the 0.001 level resulting in a very clear rank order:

$$\text{CPY} > \text{NEW} > \text{TTS}$$

In Figure 5 we present the average rank order for each of the ten sentences as well as the average over all sentences. We observe that the above rank order is achieved in 8 out of 10 sentences. For sentence 6, the NEW model is judged slightly better on the average than the CPY prosody. For sentence 7, the TTS prosody was judged to be slightly better than the NEW model.

In two cases (sentences 3 and 6) the NEW model performs close to the CPY model.

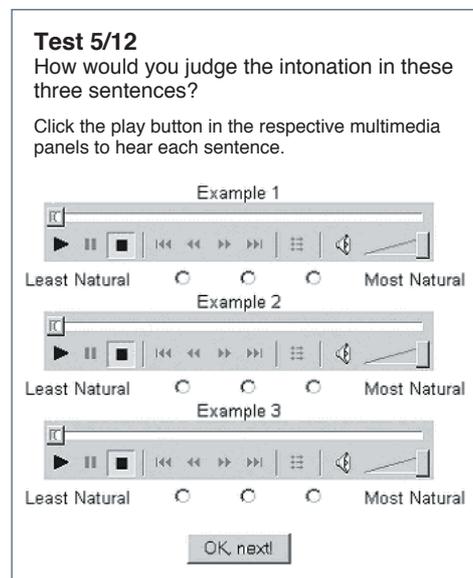


Figure 8 Interface for subjective evaluation

| | Preferred prosody model | | |
|------|-------------------------|------|------|
| | TTS | NEW | CPY |
| TTS | - | 71.4 | 83.3 |
| NEW | 28.6 | - | 76.7 |
| ORIG | 16.7 | 23.3 | - |

Table 4 Percentages of preference for one prosody model over another

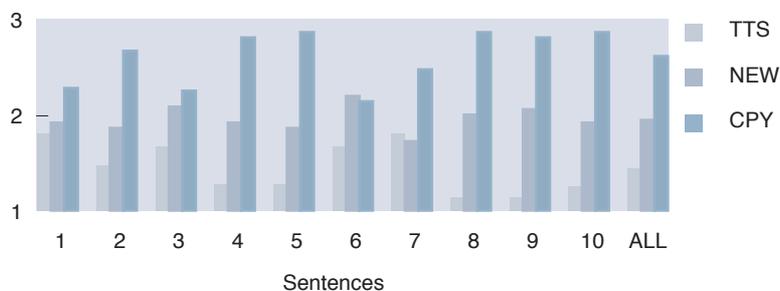


Figure 9 Average preference score for each sentence

5 Conclusions and Future Work

In this proof of concept study we have evaluated a new corpus based method for generating prosody in text-to-speech synthesis. We have shown that our proposed approach leads to a clear improvement over the rule-based model used in Talsmann TTS [11]. The main advantage of our approach is adaptability and simplicity: the prosody and speaking style of new speakers can be easily mimicked from annotated recordings with no need to develop new detailed acoustic-prosodic rules. The price paid is a reduced speech quality compared to full scale unit selection and concatenation systems. However, for some applications a quality comparable to GSM speech may be acceptable, especially when taking into account the significantly lower complexity of a diphone system.

Context dependent accent units seem to be appropriate units for concatenation of natural F0 contours in Norwegian. Still, the intonation was not judged as good as the copy synthesis and in future work it would be interesting to examine issues not covered in our study.

In this experiment we used a very simple selection procedure optimizing the cost at each concatenation point. A better solution would be to implement a Viterbi search in the graph of candidate units to select the chain of units that minimizes the cost function for the complete utterance.

One very interesting feature of our approach is the possibility to apply speaking styles adapted to specific applications, e.g. news reading, conversational style for use in dialog systems, etc. An even more interesting possibility is to implement different Norwegian dialects.

One challenge that this approach will have in common with full scale unit selection and concatenation TTS systems is design and (automatic) fast building of new databases of different speaking styles and dialects. This will require automatic detection of the relevant prosodic features of spoken Norwegian.

Rhythm is an integral part of a person's speaking style. In our experiment we trained a prediction model for the duration factor, based on the same database as we used for intonation transplantation. An interesting alternative would be to transplant rhythmic parameters from the database as well.

A further extension of the work could be to integrate the phonological prosody model as part of a segmental unit selection system along the lines presented in [16].

6 References

Sound examples for this paper can be found at: www.telenor.no/fou/prosjekter/taletek/prositt/

- 1 Bunnell, H T, Hoskins, S R, Yarrington, D. Prosodic vs. Segmental Contributions to Naturalness in a Diphone Synthesizer. In: *Proceedings of the third ESCA/COCOSDA Workshop on Speech Synthesis*, Jenolan, Australia, Nov 25–29, 1998.
- 2 Plumpe, M, Meredith, S. Which is more Important in a Concatenative Text To Speech System – Pitch, Duration or Spectral Discontinuity? In: *Proceedings of the third ESCA/COCOSDA Workshop on Speech Synthesis*, Jenolan, Australia, Nov 25–29, 1998.
- 3 Aubergé, V. Developing a structured lexicon for synthesis of prosody. *Talking Machines*. Amsterdam, Elsevier, 1992, 307–321.
- 4 Morlec, Y, Bailly, G, Aubergé, V. Synthesis and evaluation of intonation with a superposition model. *Proceedings EUROSPEECH '95*, Madrid, Sept 1995, 2043–2046.
- 5 Malfrère, F, Dutois, T, Mertens, P. Fully Automatic prosody generator for Text-To-Speech. *Proc ICSLP '98*. Sydney, Sep 1998.
- 6 Fretheim, T. Themhood, Rhemhood and Norwegian Focus Structure. In: *Folia Linguistica XXVI/1-2*. Berlin, Mouton/de Gruyter, 111–150, 1992.
- 7 Nilsen, R A. *Intonasjon i interaksjon, sentrale problemstillinger i norsk intonasjon*. Trondheim, University of Trondheim, 1993. (Dr.Art. dissertation.) (In Norwegian)
- 8 Natvig, J E, Heggveit, P O. *PROSDATA – A speech database for study of Norwegian prosody v2.0*. Kjeller, Telenor R&D, 2000. (R&D note N 20/2000.)
- 9 Heggveit, P O. *CART based duration modelling for Norwegian text-to-speech*. Kjeller, Telenor R&D, 2000. (R&D note N 19/2000)

- 10 Natvig, J E, Heggveit, P O. *Prosit – Akustisk prosodimodellering for norsk talesyntese*. Kjeller, Telenor R&D, 2000. (R&D note N 68/2000.)
- 11 *Talsmann : A text-to-speech system for Norwegian*. 2003, June 17 [online] – URL: <http://www.fou.telenor.no/prosjekter/taletek/talsmann>
- 12 Ottesen, G, Horvei, B, Stensby, S. *Modell for norsk intonasjon – Sluttrapport*. Trondheim, SINTEF-DELAB, 1992. (Rapport STF 40-F92182)
- 13 Taylor, P et al. *Edinburgh Speech Tools Library, System Documentation Edition 1.2*. Centre for Speech Technology, University of Edinburgh. 2003, June 17 [online] – URL: http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0/
- 14 *SAMPA*. 2003, June 17 [online] – URL: <http://www.phon.ucl.ac.uk/home/sampa/home.htm>
- 15 Hermes, D J, van Gestel, C. The frequency scale of speech intonation. *J. Acoust. Soc. Am.*, 90 (1), 97–102, 1991.
- 16 van Santen, J P H. Combinatorial issues in text-to-speech synthesis. In: *Proceedings of the European Conference on Speech Communication and Technology*, Rhodos, Greece, 2, 553–556, 1997.
- 17 Möbius, B. Rare events and closed domains: Two delicate concepts in speech synthesis. In: *4th ESCA Workshop on Speech Synthesis*, Scotland, 2001.
- 18 Taylor, P, Black, A W. Speech synthesis by phonological structure matching. *Proceedings of Eurospeech*, Budapest, Hungary, 2, 623–626, 1999.

Speech Centric Multimodal Interfaces for Mobile Communication Systems

KNUT KVALE, NARADA DILP WARAKAGODA AND
JAN EIKESET KNUDSEN



Knut Kvale (39) is Research Scientist at Telenor R&D and Professor II at the Stavanger University College. He graduated from the Norwegian Institute of Technology as *Siv.Ing. (MSc)* in 1987 and *Dr.Ing. (PhD)* in 1993. Knut Kvale joined the Speech Technology Group of Telenor R&D in 1994. From September 1998 he has been the Research Manager of this group.

knut.kvale@telenor.com



Narada Warakagoda (38) is a research scientist at Telenor R&D. He graduated from the Norwegian University of Science and Technology as *Siv.Ing. (MSc)* in 1994 and *Dr.Ing. (PhD)* in 2001. During his work towards the PhD degree, he developed automatic speech recognition algorithms based on the theory of nonlinear dynamical systems. After joining Telenor R&D, his research began to orientate more towards speech technology based telecom applications. Recently he has been involved in several projects on speech centric multimodal systems. His current research interests are multimodal interfaces which can exhibit human-like intelligent and emotional behaviour.

narada-dilp.warakagoda@telenor.com

Natural conversational man-machine interfaces should support multimodal interaction. Generally, a multimodal system combines natural input modes such as speech, touch, manual gestures, gaze, head and body movements, and searches for the meaning of these combined inputs. The response can be presented by a multimedia system.

The main focus of this paper is the technical aspects of implementing multimodal interfaces for mobile terminals. With the limited size and processing power of these terminals, we have restricted the functionality to *speech centric* multimodal interfaces with *two* input modes: speech (audio) and touch, and two output modes: audio and vision. That is, the input combines automatic speech recognition and a pen to click areas on the touch-screen, or pushing buttons on the small terminal. The output is either speech (synthetic or pre-recorded) or text and graphics.

1 Introduction

Since small mobile terminals have limited keypads and small screens we have to use alternative communication channels when interacting with these terminals. Speech is a natural and convenient way to express complex questions to a service. Speech is also the best option when eyes and hands are busy, e.g. when driving a car. Thus, a well-designed user interface has to support automatic speech recognition (ASR). However, when the speech recogniser fails or when it is not appropriate to speak, the user may want to select icons or scroll menus by pointing at the screen or even write the commands. Input channels such as touch sensitive screens, keyboards or keypads are therefore also required. The terminals' output channels should at least support visual (text and graphics) presentation and audio. Listening may be a slow way of obtaining information, and it is difficult to remember anything more than a few names or numbers. Screens are needed to display graphics and information that is tedious to listen to.

For some tasks it is natural and convenient to ask questions orally while pointing at related objects on the screen, e.g. when asking for the distance between two points on a map. In this case we want to combine different "senses" seamlessly in the user-interface. Multimodal interfaces combine the different input signals, extract the combined meaning from them, find requested information and present the response in the most appropriate format.

Hence, a multimodal human-computer interface (HCI) gives us the opportunity to choose the *most natural interaction* pattern. If the preferred mode fails in a certain context or task, we may switch to a more appropriate mode or we can combine modalities.

In the last two decades there has been a huge research activity within multimodal user interfaces. In 1980 Bolt [1] presented the "Put That There" concept demonstrator, which processed speech in parallel with manual pointing during object manipulation. Since then major advances have been made in speech recognition algorithms and natural language processing, in handwriting and gesture recognition, as well as in speed, processing power and memory capacity of the computers. Today's multimodal systems are capable of recognizing and combining a wide variety of signals such as speech, touch, manual gestures, gaze, head and body movements. The response can be presented by a multimedia system. These advanced systems need various sensors and cameras and a lot of processing power and memory. They are therefore best suited for kiosk applications. An overview of various multimodal systems can be found in [2], [3], [4], [5].

The work presented in this paper is based on the research work at the Speech Technology group at Telenor R&D aiming at implementing a test platform for speech-centric multimodal interaction with small mobile terminals. For applications on mobile terminals, with limited size and processing power, we have restricted the functionality to *speech centric* multimodal interfaces with two input modes: speech (audio) and touch, and *two* output modes: audio and vision. Our main focus is on exploiting multimodal speech dialogues to improve dialogue success rate, dialogue completion time and user friendliness in applications basically for 3rd generation mobile communication systems (3G/UMTS).



Jan Eikeset Knudsen (41) is Research Scientist at Telenor R&D. He graduated from the Norwegian Institute of Technology as Siv.Ing. (MSc) in 1987. Jan Eikeset Knudsen joined the Speech Technology Group at Telenor R&D in 1988.

jan-eikeset.knudsen@telenor.com

2 What is Multimodality?

2.1 Modality, Multimodal and Multimedia

The term *modality* refers to a form of sensory perception: hearing, vision, touch, taste and smell. For our research on human-machine interaction, we define modality as a communication channel between the user and the device.

The modes above can be combined in a multimodal interface, containing audio (e.g. in the form of speech), vision (in the form of text and graphics, or moving video), and touch. We do not consider services using one particular input mode, e.g. speech, and another output mode, e.g. text/graphics as multimodal services.

We distinguish between *multimode* and *multimedia*; that is, *media* is the representation format for the information or content in a certain *mode*. For example, speech and music are two media formats in the auditory mode. Text, graphics and video are examples of media types in the visual mode.

2.2 Combining Multiple Modalities

Multiple input and output modalities can be combined in several different ways. We apply the scheme proposed by the World Wide Web Consortium (W3C) [8], which distinguishes between three different ways of combining multimodal inputs and outputs: *Sequential*, *uncoordinated simultaneous* and *coordinated simultaneous* multimodal input/output, as discussed below.

2.2.1 Multiple Input Modalities

Sequential Multimodal Input

This is the simplest type, where inputs from different modalities are interpreted separately. For each dialogue state, there is only one input mode available, but in the whole interaction more than one input mode may be used.

Sequential multimodal input is often used in system-driven applications.

Uncoordinated Simultaneous Multimodal Input

In this situation several parallel input modes are active at the same time. This means that the users can choose the input mode they prefer at each dialogue stage. However, only one of the input channels is interpreted (e.g. the first input).

Coordinated Simultaneous Multimodal Input

Here, more than one input mode is available, and *all* inputs from the multiple modalities within a given time window are interpreted. The interpretation depends on the fusion of the partial information coming from the channels.

Coordinated simultaneous modalities may be the most natural way of interacting with computers, but it is by far the most complicated scenario to implement.

2.2.2 Multiple Output Modalities

W3C [8] distinguishes between three different implementation schemes for multimodal *outputs* in a similar manner. On the output side the sequential and non-coordinated simultaneous use of modes is less apparent, because the graphical display is static: it remains visible during times when speech is played (and the graphical image cannot be changed). In coordinated simultaneous multimodal output, information may be conferred by means of a spoken message that coincides with changes in the graphical display and perhaps also with gestures of an on-screen presentation agent.

2.3 Speech Centric Multimodality

Some multimodal systems apply advanced input 'devices', such as gaze tracking and facial expression recognition, and outputs e.g. facial animation in the form of human-like presentation agents on the screen as in "Quickset" [2, 3, 4, 5], "SmartKom" [9, 10], and "Adapt" [11, 12].

However, for telecommunication services on small mobile terminals, we have constrained our multimodal research issues to *speech centric multimodality* with *two* input modes: speech (audio) and touch, and two output modes: audio and vision. That is, the input combines automatic speech recognition and a pen for clicking areas on the touch-screen, or pushing buttons on the small terminal, also called "tap and talk" functionality. The output is either speech (synthetic or pre-recorded) or text and graphics.

Speech centric multimodality utilises the fact that the pen/screen and speech are *complementary*: The advantage of pen is typically the weakness of speech and vice versa. With speech it is natural to ask one question containing several key words, but it may be tedious to listen to all information read aloud – speech is inherently sequential. With pen only, it may be hard to enter data, but it is easy to get a quick overview of the information on the screen, as summarised in Table 1.

Hence, we believe that combining the pen and speech input in a speech centric multimodal interface will lead to a more efficient dialogue through better error avoidance, error correction and error recovery, as exemplified below:

- Users will select the input mode they judge to be less prone to error;

| Only pen input, screen output | Pure speech input/output |
|--|---|
| Hands and eyes busy – difficult to perform other tasks | Hands and eyes free to perform other tasks |
| Simple actions | Complex actions |
| Visual feedback | Oral feedback |
| No reference ambiguity | Reference ambiguity |
| Refer to items on screen only | Natural to refer to absent items also |
| No problem with background noise | Recognition rate degrades in noisy environments |

Table 1 Comparison between the two complementary user interfaces: Pen-only input and screen (visual) output versus a pure speech-based input/output interface

- Users tend to use simpler language which reduces the complexity of the Spoken Language Understanding (SLU) unit;
- Users tend to switch modes after system errors and thus facilitating error recovery;
- Improved error recovery by combining N-best lists and confidence scores and selection from multiple alternatives via display;
- Improved speech recognition performance by context sensitive ASR grammars with respect to other modalities, e.g. focused field in display.

In addition, the “tap and talk”-interface explicitly provides dialogue state (i.e. the tapped field), simplifying the dialogue management compared to other multimodal systems.

Microsoft’s Mipad (Multimodal Interactive Pad) [13, 14] is one example of multimodal interaction limited to speech and pen. With MiPad the user interacts with the PDA by tapping and holding on a field and uttering appropriate content to it, i.e. uncoordinated simultaneous multimodal input. MiPad is an application prototype offering

a conversational, multimodal interface to Personal Information Manager (PIM) on a PDA, including calendar, contact list and e-mail.

2.4 Speech Centric Multimodality in Form-filling Applications: An Example

In order to illustrate the concepts of speech centric multimodality, we have developed a multimodal demonstrator platform with two categories of multimodal dialogues: Form filling applications [15] and a map/location-based system. The overall architecture is described in chapter 4. In this section we exemplify the benefits of speech centric multimodality in two form filling telephony applications: A train timetable information retrieval service and a “yellow pages” service. The system architecture for form-filling applications is here basically appropriate for the *Sequential* and *Non-coordinated simultaneous* multimodal input types.

Figure 1 shows the graphic user interface (GUI) in three dialogue steps of the service for the Norwegian train timetable information retrieval application.

1 This entry page appears on the screen when the service is called up. Below the text heading: “Where do you want to go?” there are five input form fields: Arrival and departure station, date and time of arrival and the number of tickets. The questions are also read aloud by text-to-speech synthesis (TTS).

2 This screen shows the result of the user request in natural language¹⁾: “I want to go from Kristiansand to Bodø next Friday at seven o’clock”. The key words in the utterance were recognised correctly and the corresponding fields filled in, giving the user an immediate feedback on the screen. The call

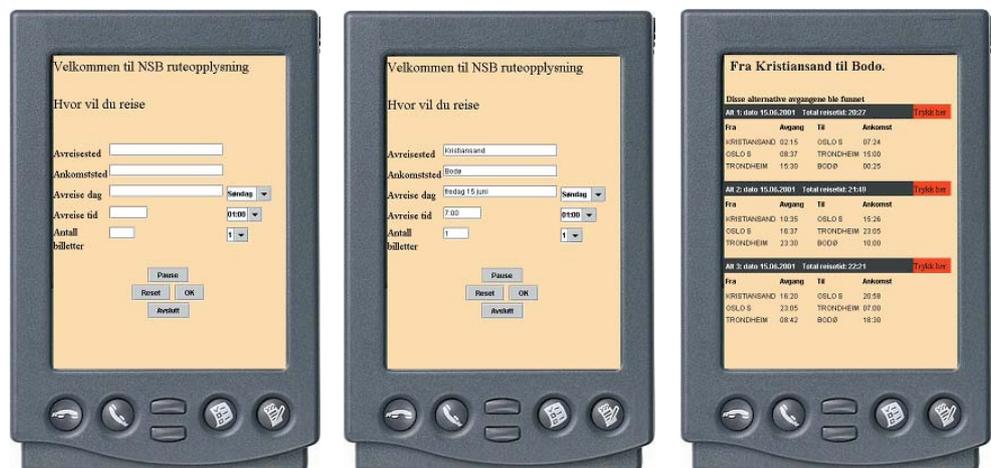


Figure 1 The GUI for the train timetable information retrieval application

¹⁾ In Norwegian: “Jeg vil reise fra Kristiansand til Bodø neste fredag klokka sju”.

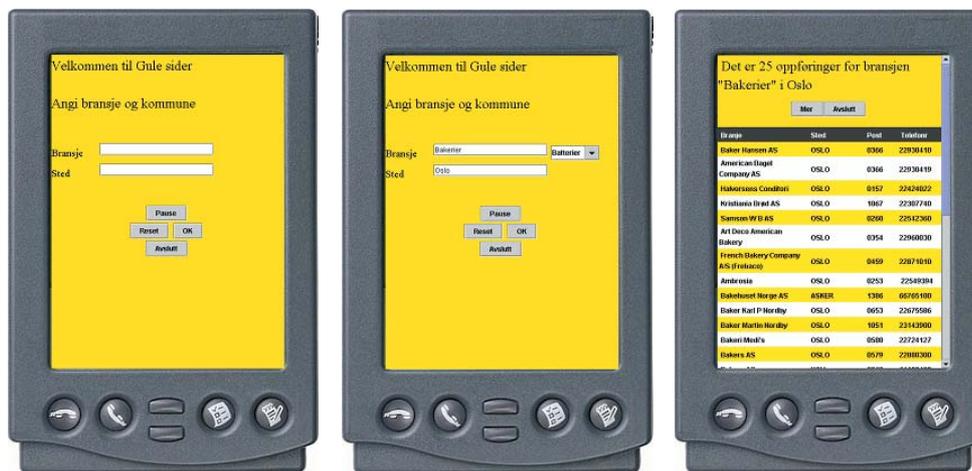


Figure 2 Yellow pages application

was made on June 10, so “next Friday” was correctly interpreted as June 15. Since all the information in the form fields on the screen is correct the user confirms by pushing the ‘OK’ button, and the system gets the requested information from the railway company web portal.

- 3 The result of the web request is presented on the screen. Usually three or four realistic alternatives are depicted on the screen. The user may then click on the preferred travel alternative, or say the alternative number. Then the dialogue goes on to ask how many tickets the customer wants for the selected trip, and the demonstrator service ends.

In the example in Figure 1, all the words were correctly recognised and understood and the visual presentation of information was much more efficient than audio feedback. Thus the customers efficiently obtained what they wanted. However, in real world speech-enabled telephony applications speech recognition errors will unavoidably occur. Correcting speech recognition errors in speech only mode (no visual feedback) is very difficult and reduces user satisfaction. But, with speech centric multimodal interface it is easier to correct ASR-errors in these form-filling services. If some of the information on the screen is wrong, the user corrects it by clicking on the field containing the erroneous words and then either saying the correct word once more or tapping on the correct word from the N-best list, which occurs on the right hand side of the field.

Figure 2 illustrates this situation in a “yellow pages” application:

- 1 On the entry page that appears on the screen when the service is called up and under the

text heading “Welcome to Yellow pages”, there are two input form fields: Business sector and municipal (Norwegian: “Bransje” and “sted”).

- 2 When the user asked in natural language²⁾ “I want bakeries in Oslo”, the ASR recognised the key words in the utterance and filled in the corresponding fields, giving the user an immediate feedback on the screen. Note that the N-best list on the right hand side of the sector field contains the alternative “Batteries”, i.e. the word “batteries” has the second best confidence score. Since all the information in the form fields on the screen is correct the user pushes the ‘OK’ button, and the system gets the requested information from the service provider.
- 3 The requested information is displayed on the screen. There are 25 bakeries in this listing, which would have been rather tedious to listen to. Here, the user easily gets a quick overview and clicks on the preferred baker.

The actions and benefits of multimodality in the form filling examples are summarised in Table 2.

3 General Implementation Aspects

The generic multimodal dialogue system shown in Figure 3 with several input and output channels, raises several important issues when implementing multimodal architectures, such as:

- Selection of suitable software architecture for implementing the logical elements shown in Figure 3 in a real system;
- Funnelling of multiple channels through the heart of the system, namely the dialogue manager.

²⁾ Pronounced in Norwegian as: “Jeg vil gjerne ha bakerier i Oslo”. Here the business sector (Norwegian: “Bransje”) is “bakerier” and the municipal (Norwegian “sted”) is Oslo.

| User actions | Benefits of multimodality |
|---|---|
| Natural language input, asking for several different pieces of information in one sentence. | Speech is most natural for asking this type of question. Speech is much faster than typing and faster than selecting from a hierarchical menu. |
| Reads information shown on the screen. | The user gets a quick overview – much quicker than with aural feedback reading sentence by sentence. |
| Taps in the field where the ASR-error occurs, and taps at the correct alternative in the N-best list. | Much easier to correct ASR-errors or understand rejections than in complicated speech-only dialogues. Better error control and disambiguation strategies (e.g. when there are multiple matching listings for the user query). |

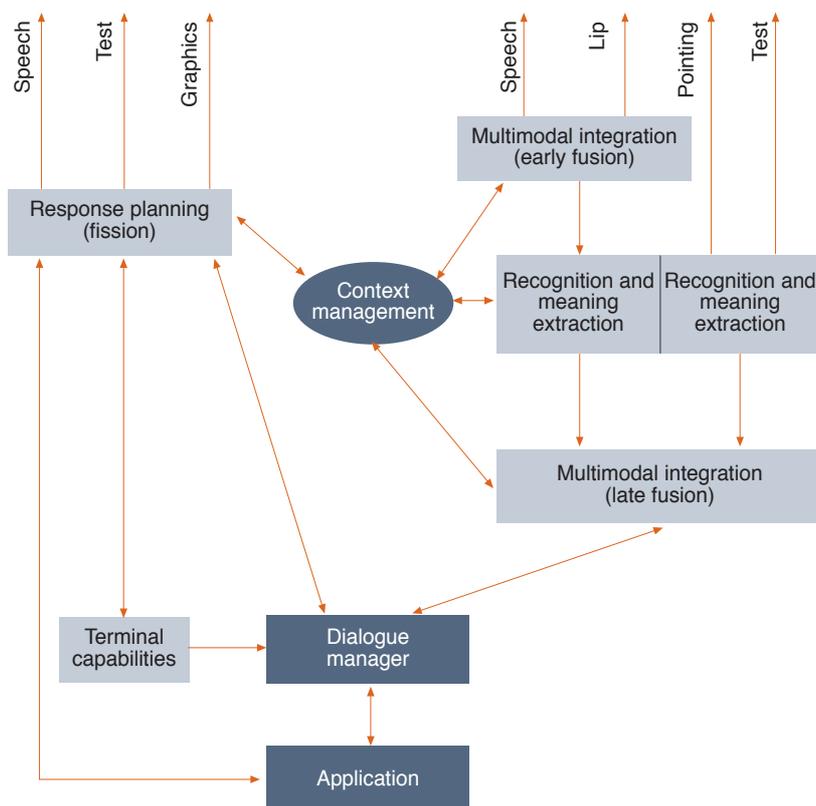
Table 2 Benefits of multimodality in the form-filling applications

3.1 Distributed Processing

In advanced multimodal systems, several input/output channels can be active simultaneously. In order to cope with these kinds of multimodality, an architectural support for simultaneous information flow is necessary. Furthermore, it is desirable to run different functional modules separately (often on different machines), in order to deal more effectively with the system's complexity. The so-called *distributed processing paradigm* matches these requirements quite nicely, and therefore most of the multimodal system architectures are based on this paradigm.

There are many different approaches to implementing a distributed software system. One can

Figure 3 A generic multimodal dialog system architecture



for example choose a very low level *socket*-based approach or one of the higher level approaches such as Parallel Virtual Machine (PVM) [42], Message Passing Interface (MPI) [43], RPC-XML [44], and SOAP [45]. On an even higher level, object oriented distributed processing approaches such as CORBA, DCOM, JINI and RMI are available. However, the most attractive approach to implementing multimodal systems is based on *co-operative software agents*. They represent a very high level of abstraction of distributed processing and offer a very flexible communication interface.

3.2 Existing Agent-Based Platforms

There are several well-known agent architectures that have been used to build multimodal systems, such as GALAXY Communicator (a public domain reference version of DARPA Communicator) maintained by MITRE [19], Open Agent Architecture (OAA) from SRI international [20] and Adaptive Agent Architecture (AAA) from Oregon Graduate Institute (OGI) [21]. In these architectures a set of specialised agents are employed to perform different tasks. Two given agents can communicate (indirectly) with each other through a special agent called a *facilitator*. Usually, this inter-agent communication is a *service request* and a *response* to such a request. The facilitator performs matchmaking between a service provider and a requester.

We found that GALAXY Communicator is the most suitable agent-based platform for our purpose. A detailed description is given in section 4.2.1. In this system, messages sent between agents and the central facilitator called a *hub*, are based on a simple attribute-value type data structure. This *hub-spoke* type architecture allows easier asynchronous and simultaneous message exchange than for example a *serial* architecture does. In addition to a message passing between the hub and the agents, direct communication between any two agents is also possible. Such a direct connection is useful when large amounts of data, such as audio, have to be efficiently passed between two agents. One drawback with GALAXY Communicator, however, is its dependency on a single facilitator, whose failure will cause a complete system breakdown. In AAA this problem has been addressed by introducing many facilitators.

3.3 Fusion and Fission

Since an advanced multimodal system such as the one shown in Figure 3 has more than one input and/or output channel, there must be mechanisms to map:

- Several input channels to a single semantic stream, i.e. *fusion*;

- A single semantic stream to several output channels, i.e. *fission*.

From a technical point of view, fusion, also called *multimodal integration*, deserves a higher attention than fission, because a good fusion strategy can help reduce the recognition errors.

Usually, fusion is divided into two classes, *early fusion* and *late fusion*. Early fusion means integration of the input channels at an early stage of processing. Often this means integration of feature vectors, before they are sent through the recogniser(s). Late fusion means integration of the recogniser outputs, usually at a semantic interpretation level. Late fusion seems to have attracted more interest than early fusion, probably because it only needs the recogniser outputs and no changes of existing modules (such as feature extractors, recognisers).

In one of its simplest forms, late fusion can be performed by simple table look-ups. For example, assume that we have two input channels. Then we can maintain a (two-dimensional) table, where rows and columns correspond to alternative outcomes of the recognisers acting on channel 1 and channel 2 respectively. Each cell of the table can be marked 1 or 0, indicating whether this particular corresponding combination is valid or invalid. Then the fusion procedure for a given pair of recogniser output lists would be to scan the (recogniser) output combinations in the decreasing order of likelihood and find the first valid combination by consulting the table.

The above procedure can be extended to handle uncertainty associated with recognition by considering joint probability of the recogniser outputs from the two channels. One simple approach for computing these joint probabilities is to assume that two recognition streams are statistically independent. However, the fusion performance (i.e. multimodal recognition performance) can be enhanced by dropping this assumption in favour of more realistic assumptions [22].

Table look-up based fusion is not very convenient when the semantic information to be integrated is complicated. In such cases *typed feature structures* can be used. This data structure can be considered as an extended, recursive version of *attribute-value* type data structures, where a value can in turn be a feature structure. Typed feature structures can be used for representing meaning as well as fusion rules. Integration of two or several feature structures can be achieved through a widely studied algorithm called *feature-structure unification* [2].

In fusion, temporal relationships between different input channels are very important. In multi-

modal systems this issue is usually known as *synchronization*. In most of the reported systems in the literature, synchronization is achieved by considering all input contents that lie within a pre-defined time window. This can be done by employing timers and relying on the real arriving times of the input signals to the module responsible for performing fusion. However, a more accurate synchronization can be obtained by time-stamping all inputs as soon as they are generated since this approach will remove the errors due to transit delays. Note, however, that input synchronization is meaningful only for coordinated multimodality.

3.4 Dialogue Management

A dialogue manager is usually modelled as a finite state machine (FSM), where a given state S_t represents the *current context*. One problem with this modelling approach is the potentially large number of states even for a relatively simple application. This can be brought to a fairly controllable level by considering a hierarchical structure. In such a structure there are only a few states at the top level. But each of these states is thought to be consisting of several substates that lie on the next level. This can go on until the model is powerful enough to describe the application concerned.

When the user generates an event, a *state transition* can occur in the FSM describing the dialogue. The route of the transition is dependent upon the input. That means that state transition is defined by the tuple (S_t, I_t) , where S_t is the current state and I_t is the current user input. Each state transition has a well-defined end state S_{t+1} and an output O_t . In other words the building-block-operation of the dialogue manager is as follows:

- 1 Wait for input (I_t).
- 2 Act according to (S_t, I_t) for example by looking up a database and getting the result (R_t)
- 3 Generate the output according to (S_t, I_t, R_t)
- 4 Set next state S_{t+1} according to (S_t, I_t)

The user input (I_t) is a vector which is a representation of the structure called concept table. This structure consists of an array of *concepts* and the values of each of these concepts. For example in a travel planning dialogue system the concept table can look as shown overleaf.

The column “value” of the concept table is filled using the values output by the speech recogniser and the other recognisers operating on the input modalities (e.g. a GUI tap recogniser). During the filling operation, input ambiguities can be resolved completing late fusion. Once filled, the

| Concept | Value |
|------------------|-----------|
| <FROM_CITY> | Oslo |
| <TO_CITY> | Amsterdam |
| <DEPARTURE_TIME> | 1600 |

concept table defines the current input I_t . More specifically, if the values in the concept table are $I_t(1), I_t(2), \dots, I_t(n)$, then the N-tuple $(I_t(1), I_t(2), \dots, I_t(n))$ is the current input I_t . The number of different inputs can be prohibitively large, even if the length of the concept table (M) and the number of values a given concept can take (K) is moderate. This implies that a given state in the dialogue FSM has a large number of possible transitions.

A possible remedy for this problem is to employ a clever many-to-one mapping from the original input space to a new smaller sized input space, which exploits the fact that there are many don't-care concept values.

4 An Implementation of a Multimodal Demonstrator Platform

This chapter describes the architecture and the system aspects of the multimodal demonstrator platform developed in the EURESCOM project MUST [7], [16], [17], [18]. The main purpose of the demonstrator was to implement an experimental multimodal service in order to study and evaluate the interaction with and the appreciation of such a service by real "naïve" users. The service is a map application, i.e. a tourist guide for Paris.

4.1 System Overview

The multimodal demonstrator platform consists of a relatively complex server and a thin client.

The overall architecture of the platform is shown in Figure 4.

The Application Server consists of five main autonomous modules (or servers) that inter-communicate via a central facilitator module (Hub). The modules are:

- *Voice Server* – comprises speech technological components such as Automatic Speech Recognition (ASR), Text to Speech Synthesis (TTS) and telephony (PHN) for the speech modality.
- *GUI Server* – is the gateway between the GUI Client and the server side.
- *Multimodal Server* – performs preliminary steps of multimodal integration of the incoming signals (fusion) and distributes the service response through the output channels (fission).
- *Map Server* – acts as a proxy interface to the map database.
- *Dialogue & Context Manager Server* – performs the dialogue and context management.
- *Hub* – manages the inter-communication for the modules at the server side.

The client side, or the multimodal mobile terminal, consists of two modules: The Voice Client which is a plain mobile telephone used for the voice modality, and the GUI Client which is a programmable Pocket PC (or a Personal Digital Assistant – PDA) with touch sensitive screen used for the tap and graphical modality.

All the modules in the MUST Application Server are described in section 4.2. The client side of the MUST demonstrator is described in section 4.3.

4.2 The Server Side

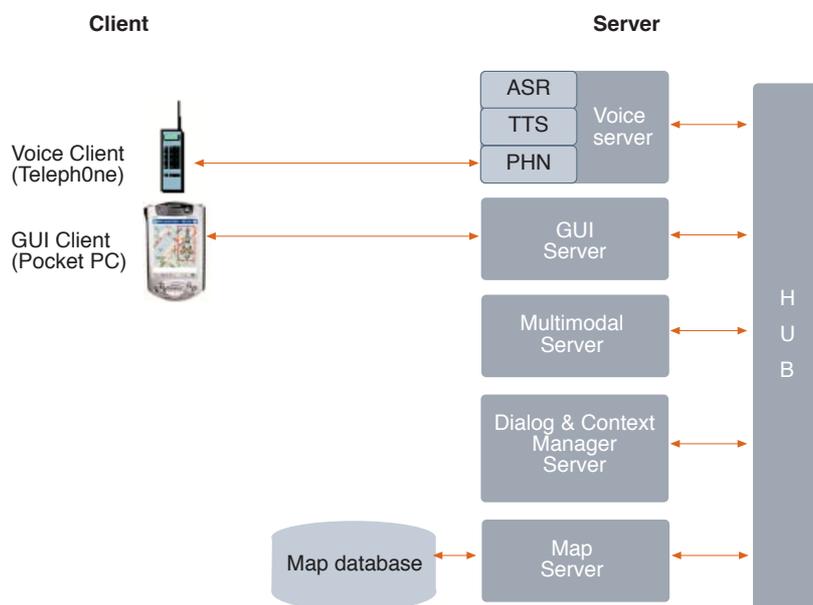
This section describes all the modules that comprise the Application Server. All the modules are implemented in Java, except for the Voice Server that is implemented in C++.

4.2.1 The Galaxy Hub

The modules communicate asynchronously by messages passing through a Hub. The Hub distributes the messages according to a set of rules in accordance with the service logic.

The Galaxy Communicator Software Infrastructure mentioned in section 3.2 was selected as the underlying software platform, which provides the Hub in Figure 4. The main features of this framework are modularity, distribution, seamless integration of the modules and flexibility in

Figure 4 The architecture of the multimodal demonstrator platform



terms of inter-module data exchange, i.e. synchronous and asynchronous communication through the Hub and directly between modules.

In addition to the Galaxy Communicator, two other promising software multimodal platforms have been considered: ATLAS [11, 12], and Smartkom [9]. The lack of availability of these software platforms in terms of licensing clarifications as well as available documentation in English led us to discard these two alternatives at an early stage and go for the Galaxy Communicator.

The Galaxy software infrastructure allows us to connect the modules (i.e. the GUI Server, Voice Server, Multimodal Server, Dialogue Server and the Map Server) together in different ways by providing extensive facilities for messages passing between the modules. A module, or a server, can very easily invoke a functionality that is being provided by another module without knowing which module that provides it or where it is running. This is an important feature of the modularity, distribution, and seamless integration.

Normally, a script is used to control the processing in the Hub, but it can also act as an autonomous facilitator in an agent-based environment. In our platform, we have followed the approach of script-based Hub control.

Galaxy messages that are passed between the modules are based on the key-value pair (attribute-value pair or *name* plus a *value*) format. This message format allows messages to encompass simple data structures that are sufficient to support less complex operations such as connection set-up, synchronisation, and disconnection. However, more complex operations such as database lookup results and GUI display information involve more complex data structures, necessitating an extension to the message format. Fortunately, the Galaxy message format is a framework that is flexible and extensible.

In the MUST project, an XML-based mark-up language (named MxML – ‘MUST eXtensible Mark-up Language’) was defined to cope with complex data structures in the Application Server. Complex data structures are represented by MxML strings and embedded in the basic key-value pair based Galaxy messages. In this way we can take the advantage of the flexible message passing mechanism provided by Galaxy Communicator and the power of XML.

The highly modular and seamless architecture of the Application Server gives a flexible system where modules easily can be removed, added or replaced. For example, we have used two alter-

native versions of the Voice Server (see section 4.2.2), and one can replace the other whenever necessary.

Due to the distributed nature of the architecture, many different deployment configurations are possible for the MUST demonstrator. We typically use two PCs – both running Windows 2000; one PC for the Voice Server (including a lot of firmware such as telephony drivers, ASR and TTS engines), and another PC for the other Servers. The demonstrator also supports a portable stand-alone version where all the modules (including the whole Voice Server) are running on one machine (i.e. a laptop).

4.2.2 Voice Server

The Voice Server (or Voice Platform) is the module that handles the voice modality. This module is built upon underlying speech technology such as ASR, TTS and telephony, and basically provides the API for these technology resources.

Because two of the partners in the MUST project (Portugal Telecom and Telenor) wished to use their own voice platforms, two different versions of the Voice Server have been developed for the MUST demonstrator. One is based on the ‘InoVox’ [23] voice platform from Portugal Telecom Inovação, and the other version is based on the ‘Tabulib’ voice platform [24] from Telenor R&D.

The Tabulib platform is currently freely available to the general public. Tabulib basically supports ISDN telephony for the input/output voice to the system, but also a proprietary VoIP (Voice over IP) solution is implemented for the development of a portable demonstrator platform without the need for a wired telephone line. The speech recogniser supported is Philips Speech-Pearl 2000 [25] recogniser. For the Text-to-Speech Synthesis, Tabulib supports the Microsoft SAPI 4.0 [26] standard, and hence, all Text-to-Speech Synthesis engines supporting this standard can be used with Tabulib. We have used a TTS engine from Microsoft for English [27] and Telenor’s TTS engine Talsmann [28] for Norwegian.

4.2.3 GUI Server

The GUI Server is the ‘gateway’ between the GUI Client (i.e. the GUI part of the multimodal mobile terminal) and the Application Server. Received data from the GUI Client is packed into Galaxy frames and transmitted to the Dialogue Server via the Multimodal Server. Data from the Dialogue Server is extracted from the received Galaxy frames, and an HTML file is generated. The generated HTML file is actually stored on an HTTP (Web) server and further

fetched by the GUI Client – which is a sort of Web browser.

The data from the Dialogue Server is in XML format and contains the contents to be presented on the GUI Client; that is, raw information such as text and images to be displayed, and coordinates for items – e.g. points of interest on a map. We use XSLT [29] in order to transform the XML body to an HTML file. It is the XSL style sheet that really defines the format of the GUI, such as the size of text fields, font types, background colours, the width of borders and list boxes. With the use of style sheets, the appearance of the GUI display can be easily altered on the fly in services where the GUI format should be dependent on the dialogue context or the user's profile. New applications can be implemented without any software upgrade on the GUI Client, since the GUI is defined by the XSL style sheets and the content in the XML body.

4.2.4 Multimodal Server

The MUST project aimed to investigate the implementation and human factor issues related to coordinated simultaneous multimodal inputs, i.e. all parallel inputs must be interpreted in combination, depending on the fusion of the information from all channels. The so-called “late fusion” approach was adopted, where the recogniser outputs are combined at a semantic interpretation level (by the Dialogue & Context Manager Server).

The Multimodal Server performs the preliminary steps of the integration of the incoming signals (pen click and voice). The temporal relationship between different input channels is obtained by considering all input signals within a pre-defined time window. The duration of this time window is a variable parameter that can be adjusted according to the dialogue state. If two input signals occur within the fusion window, then the Dialogue Server will combine the two inputs to determine the ‘meaning’ from the user, and act accordingly (late fusion). A drawback with this solution is that pen taps (e.g. on menu icons as “home” and “help”), which should not be combined with voice input, are also delayed through the fusion window. This slows down the system response.

In this version of the Must demonstrator the Dialogue Server performs the (early) fission, and the output modalities are forwarded to the destination channels.

4.2.5 Dialogue Server

The Dialogue Server, or the Dialogue and Context Manager module, can be seen as the mediator between the user on one side and the source of information on the other side. It is the Dia-

logue Server that controls the logic for the service and can be considered as the central processing unit of the system. The main tasks of the Dialogue Server is:

- Process incoming messages with textual or abstract representations from the user interactions
- Combine the multimodal inputs (late fusion)
- Context processing
- Implement the service logic rules
- Access data repository
- Handle the output information to the user (early fission)
- Error handling

The application in the MUST project (the tourist guide for Paris) is fully user driven, and it is the user who always takes the initiative in the dialogue. A big challenge in the implementation of such a dialogue system is that the context or dialogue states to a certain extent are submitted to the user's premises. The system should always be in the ready state for serving the user, i.e. receiving queries from the user at any time, which complicates the control and processing of the multimodal inputs and outputs. For system driven dialogues such as form filling applications, these implementation issues are alleviated since the Dialogue Manager takes the initiative and control in the interaction with the user.

The requests from the user are represented in terms of textual or abstract symbols by the boundary modules (here: Voice Server and GUI Server) that handle the interaction with the user. The Dialogue Server combines and processes the inputs (late fusion), and acts accordingly to fulfil the user's request (typically a query to a database). The feedback is sent to the boundary modules via the Multimodal Server in order to be presented in different modalities on the multimodal terminal (early fission).

4.2.6 Map Server

The Map Data Server is the module that receives and processes requests coming from the Dialogue Server. These queries are Galaxy messages constructed using a proprietary protocol defined to code domain information requests. These messages are parsed, interpreted and mapped to the corresponding SQL queries, which are executed to gather the requested data from the database. The data from the database is packed into a Galaxy frame, and forwarded to the Dialogue Server. An error frame is sent to

the Dialog Server if no matching data is found in the database.

4.3 The Client Side

At the time of the service specification and implementation, there were no commercial terminals available with support for the features and functionality needed for a small multimodal wireless terminal. The prototypes developed or under development in other projects were not available for third party entities.

The solution that was found to overcome this situation is the simulation of a multimodal terminal, which was done by combining a mobile GSM phone and a Pocket PC with touch sensitive screen.

4.3.1 The GUI Client

The GUI Client is implemented on a Compaq iPAQ Pocket PC running Microsoft CE 3.0/2002 (see Figure 5). The GUI Client is connected to the GUI Server via WLAN using a TCP socket connection to convey the GUI signals (tap and graphical information) back and forth between the client and server.

The GUI Client software is developed using Microsoft eMbedded Visual C++, and the main features are based on the Pocket Internet Explorer (web browser) technology. The use of ActiveX [30] controls inside the web browser gives a powerful interface that supports a variety of GUI components, such as gif image display, hotspots, push buttons, list boxes (select lists) and text fields.

The input to the GUI Client (from the GUI Server) is an HTML file. The GUI is defined and controlled by the use of Microsoft JScript [31] in the HTML code. In the MUST project, we have defined a set of JScript functions or methods for the creation of the different GUI components. This provides a powerful and flexible GUI, and it is the Application Server (here: Dialogue Manager Server and GUI Server) that defines the appearance of the GUI, and therefore no software updates are necessary on the Pocket PC in order to change the GUI. This gives a system for quick and easy design of multimodal applications.

4.3.2 The Voice Client

The speech part is for the time being handled in parallel by a mobile telephone. The users will not notice this “two terminal” solution, since the phone is hidden and the interface is transparent. Only the headset (microphone and earphones) with Bluetooth connection will be visible to the user. The headset frees the user’s hands to hold the Pocket PC and to navigate the map application with the pen giving the “illusion” that the

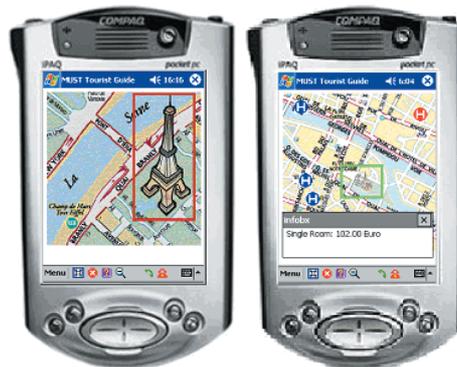


Figure 5 The GUI Client implemented on a Compaq iPAQ Pocket PC

user is interacting with a small multimodal wireless terminal.

We have also implemented a VoIP solution for the speech part. Here we utilise the audio device on the Pocket PC. The input voice is recorded on the Pocket PC, and forwarded to the Voice Server via the WLAN connection. Likewise, the audio from the Voice Server is transmitted over the same connection, and played on the Pocket PC. The advantage with such a solution is that both the GUI and speech part are implemented on one single device, giving a better “illusion” that the users are interacting with a multimodal mobile telephone. Besides, there is no need for a telephone line for the Voice Server, and the whole demonstrator can be presented everywhere as a stand-alone device. Unfortunately, this solution is for the time being not stable, due to slow CPU performance and some problems with the audio device on the current Pocket PCs.

5 When Will Multimodal Applications Take Off in the Mobile Market?

We have discussed various aspects of multimodal interfaces for mobile communications and carefully described the technical implementation of a multimodal demonstrator platform. However, when will the multimodal services take off in the mobile market? In this chapter we look at the situation today, and address some issues that are important for deployment of regular multimodal services in the near future.

5.1 Status – Existing Multimodal Mobile Services and Platforms

Today a few commercial multimodal mobile services for GSM, GPRS and WLAN are based on a combination of existing standards such as WML (WAP), HTML, VoiceXML, SMS and MMS, or *proprietary* or *de facto* standards. The multimodal interaction is in most cases limited to one modality at a time for the input and output signals. Many of the services are expensive in use, and some involve more than one terminal for the end user, e.g. a cellular phone for the speech modality, and a Personal Digital Assis-

tant (PDA) for the graphical modality. Most often, only one network operator offers the services, i.e. the one that possesses the standard, and this limits the geographical coverage and the potential of offering a mass marketing service. Below are listed some typical examples of commercial services:

- PlatiNet in Israel [32] offers services to operators such as voice-enabled chat and conferencing applications, location-based applications such as contacting the nearest taxi service or finding a hotel, and the addition of voice menus to text-based Yellow Pages, allowing easy access to any listing.
- SFR of France [33] offers text or voice coverage of events such as the football World Cup and the Tour de France, and access to other location-based services such as sports, weather, festival details, and news. The service runs on a platform from NMS Communications [34].
- Hutchison Max Telecom and Orange in India offer a multimodal Football World Cup service, which delivers audio action replays from live matches to cellular subscribers. An SMS alert is sent to subscribers at the start of a match, and they are given a number to call for live voice coverage or text updates. The service runs on the multimodal platform from OnMobile Systems, Inc. [35] in Orange's GSM network.
- Kirusa [36] and SpeechWorks [37] have established a trial multimodal service on the French mobile operator Bouygues Telecom's GPRS mobile networks in France. The service supports simultaneous multimodality, where voice and visual information seamlessly interact.

5.2 Standardization

In the past, we have seen the appearance – and disappearance – of services based on *proprietary* or *de facto* standards, for example the so-called smart phones, portable personal assistants, email devices and many more. International standardization within the telecommunication sector is definitely an important basis for large-scale regular telecommunication services in the long term. Standards provide interoperability across different networks, and standard tools for the development and deployment of the service. Standards are also highly pertinent to the forthcoming multimodal services, and the initiation of standardization activities is needed in the field of multimodal communication, particularly for the next generation mobile network (3G/UMTS).

However, due to the recent dramatic changes of the economic environment in the telecommuni-

cation sector, many telecommunication companies are forced to seek new models to generate new revenues from the existing mobile networks, e.g. GSM and GPRS, due to the large investments. Moreover, the planned 3G/UMTS mobile network is considerably delayed for a variety of reasons, from technical problems to unpredicted large investment costs. This situation will, at least for a while, slow down the standardisation effort in the field of multimodal communication. So, what one can expect in the near future is a variety of multimodal services based on a mixture of existing standards over the existing mobile networks.

We will address two important fields where standardisation is important for the forthcoming multimodal services: the standardisation of programming languages for the implementation of service logic (e.g. multimodal dialogues), and the quality and protocols in the mobile networks.

5.2.1 Programming Languages for Developing Multimodal Dialogues

The big advantage with standard languages for the development of multimodal dialogues is that the service and third party content providers can, in a convenient and flexible way, deploy new multimodal services by exploiting common development resources. In the case of cross-platform portability, services can be distributed to different application servers. A common standard will definitely speed up the dissemination and availability of such services. A good example of this is the *VoiceXML* [38], which is an XML-based mark-up language for pure voice-driven dialogues, adopted by the W3C consortium in 2000.

As regards programming languages for multimodal dialogues, W3C already in 2000 announced plans for a multi-modal mark-up language [39], but little public documentation has been available since then. An interesting activity within W3C is the XHTML+Voice profile [40] basically designed for Web clients that support visual and spoken interaction.

In July 2002 the SALT Forum [41], backed by industry heavyweights, released version 1.0 specification of the SALT mark-up language. SALT is an extension of HTML and other mark-up languages (e.g. cHTML, XHTML and WML), which basically add a speech interface to web applications and services, for voice only (e.g. telephony) and for multimodal browsers.

Surely, the ongoing activities within W3C and the recent announcement of the SALT Forum are a significant attempt at standardising implementation languages for multimodal dialogues. However, most of these languages are focusing

on web applications with dialogues of the form-filling nature.

For more advanced multimodal dialogues, with mixed initiative (both user and system driven) dialogues, and with support for all the three types of multimodal inputs (i.e. *sequential*, *uncoordinated simultaneous* and *coordinated simultaneous*), we believe that there are still some challenges to overcome before a general standard for multimodal dialogues can be drafted.

5.2.2 Mobile Wireless Networks – Quality and Protocols

The new forthcoming multimedia and multimodal mobile services will certainly put some requirements on the mobile networks. Key parameters are bandwidth, transmission delay, simultaneous and synchronous transmission of voice and data, handover capabilities, and quality of service. Of course, such parameters are to a certain extent given for the existing mobile communication networks and the forthcoming 3G/UMTS, and one should rather estimate the opportunity as well as the limitations for deploying multimedia services in mobile networks.

Critical network scenarios are the ones that involve handover between different network segments (e.g. between GPRS and WLAN) with different properties. It is important that the quality of the connection is stable, i.e. parameters such as bandwidth, jitter and time lag should, to the extent possible, be stable throughout the handover operation. Today handover in the existing mobile networks is often a problem, especially for real time traffic such as voice and video.

Standard protocols for the network transmission are an important issue. There is a need to define multi-channel transmission protocols for the different multimedia or multimodal communications, from the simplest including only voice and pen click, to more advanced applications including other modalities such as gesture. The protocols are required to work across different types of mobile networks obtaining a transparent connection. The development of such protocols should be done within standardisation bodies such as ITU, ETSI, 3GPP and IETF.

5.3 Multimodal Mobile Terminals

Widespread use of mobile multimodal services requires suitable terminals. Technically, we can classify mobile terminals into two main classes: Pre-programmed terminals and factory and programmable terminals.

Pre-programmed terminals need to be compatible with the prevailing multimodal standards, at both application and communication levels.

However, standardisation of multimodal protocols is still in its infancy and it may take some time before such standard compliant multimodal services will be taken into large-scale use.

Programmable terminals, on the other hand, allow swift introduction of multimodal services by using proprietary protocols. However, multimodal services require terminals that can be used with a network supporting simultaneous voice and data transmission at a sufficiently high bit rate. The emerging mobile networks, GPRS and UMTS, as well as WLAN available in hotspots, definitely satisfy this requirement, because they inherently support the IP protocol. Even the existing GSM network can be used by employing a modem to simulate an IP layer on top of the circuit switched net. In order to enable the use of proprietary protocols, the service provider first has to release a client program that the users can install in their terminals. This client and the service can then communicate with each other using their own application protocol.

There is already a large number of programmable mobile terminals on the market which are potential candidates for use with multimodal services. Compaq iPAQ, Toshiba Pocket PC, HP Jornada, O2 xda and Siemens PocketLoox etc. are some of the popular ones. They all come with the pre-installed Microsoft Windows CE/Pocket PC operating system for which a complete programming environment is available. These terminals have relatively large colour screens, 200 MHz or more CPU frequency, 32 MB or more RAM, audio ports and PCMCIA/CF ports. Except for O2 xda, they do not come with built-in communication hardware. However, the PCMCIA or CF (Compact Flash) port allows insertion of a communication module, such as a GPRS or WLAN card. Siemens has announced that a specially designed GPRS module for PocketLOOX will be available this year. There is reason to believe that this trend will continue and even more complete communication terminals of this kind will be available in the near future.

In spite of these developments, mobile terminals cannot be expected to be so resource-rich that sophisticated components of multimodal systems (such as speech recognition and text-to-speech synthesis modules) can be contained by the terminal itself. However, this is not an obstacle to the provision of multimodal services, as speech engines and other resource hungry components can be placed in the network. But this has significant architectural and business implications for the mobile operator, application provider, and end user, and understanding this is perhaps more important than it appears.

Even though the mobile terminals seem to have overcome the major technological barriers on the way towards multimodal services, the price of these terminals is still high. Generally a mobile terminal capable of delivering multimodal services of reasonable quality is at least ten times more expensive than a usual mobile telephone. Therefore, one cannot expect mobile multimodal services taking off until the price comes down to a level that the consumers will accept.

Human Factors

While all these engineering and technical improvements are important, the ultimate success of a multimodal system has to be assessed from the user's viewpoint. Understanding the nature and constraints of the new access interfaces is critical for commercial success of future services on mobile terminals. Also, it is important to understand why people opt for certain (combinations of) modalities in the given context, in order to develop efficient multimodal interfaces and services [3]. It is therefore necessary to run user experiments and quantify the advantages of the multimodal system in terms of user-satisfaction, dialogue success rate and dialogue completion time. These experiments are high on our agenda.

Acknowledgements

In this research activity, Telenor R&D has collaborated with France Telecom and Portugal Telecom and researchers from the Netherlands in the EURESCOM³⁾ project called "MUST – Multimodal, multilingual information Services for small mobile Terminals" [6], [7].

We would like to thank our colleagues in the Speech Technology Group at Telenor R&D, and our colleagues in the MUST project for valuable discussions and cooperation.

References

- 1 Bolt, R. Put That There : Voice and Gesture at the Graphics Interface. *Computer Graphics*, 14 (3), 262–270, 1980.
- 2 Oviatt, S et al. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *Human Computer Interaction*, 15 (4), 263–322, 2000.
- 3 Oviatt, S. Ten Myths of Multimodal Interaction. *Communications of the ACM*, 42 (11), 74–81, 1999.
- 4 Oviatt, S. Multimodal interface research : A science without borders. *Proc. 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, III, 1–4, 2000.
- 5 *Center for Human-Computer Communication*. 2002, November 8 [online] – URL: <http://www.cse.ogi.edu/CHCC/index.html>
- 6 *Eurescom*. 2002, November 8 [online] – URL: <http://www.eurescom.de/>
- 7 *MUST – Multimodal and Multilingual Services for Small Mobile Terminals*. Heidelberg, EURESCOM Brochure Series, May 2002. (<http://www.eurescom.de/public/projects/P1100-series/P1104/default.asp>)
- 8 *W3C – Multimodal Requirements for Voice Markup Languages*. 2002, November 8 [online] – URL: <http://www.w3.org/TR/multimodal-reqs>
- 9 *SMARTCOM – Dialog-based Human-Technology Interaction by Coordinated Analysis and Generation of Multiple Modalities*. 2002, November 8 [online] – URL: http://www.smartkom.org/start_en.html
- 10 Wahlster, W et al. SmartKom: Multimodal Communication with a Life-Like Character. *EUROSPEECH-2001*, Aalborg, Denmark, 1547–1550, 2001.
- 11 Gustafson, J et al. Adapt – A Multimodal Conversational Dialogue System In An Apartment Domain. *Proc. 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, II, 134–137, 2000.
- 12 Beskow, J et al. Specification and Realisation of Multimodal Output in Dialogue Systems. *Proc. 7th International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, USA, 181–184, 2002.
- 13 Huang, X et al. MiPad : A multimodal interaction prototype. *Proc. ICASSP 2001*.
- 14 Wang, Ye-Yi. Robust language understanding in MiPad. *Proc. EUROSPEECH-2001*, Aalborg, Denmark, 1555–1558, 2001.
- 15 Kvale, K, Warakagoda, N D, Knudsen, J E. Speech-centric multimodal interaction with

³⁾ EURESCOM, the European Institute for Research and Strategic Studies in Telecommunications, performs collaborative R&D in telecommunications. The EURESCOM shareholders are mainly the major European network operators and service providers.

- small mobile terminals. *Proc. Norsk Symposium i Signalbehandling (NORSIG 2001)*, Trondheim, Norway, 12–17, 2001
- 16 Boves, L, den Os, E (eds.). *Multimodal services – a MUST for UMTS*. January 2002. 2002, November 8 [online] – URL: <http://www.eurescom.de/public/projectresults/P1100-series/P1104-D1.asp>
 - 17 Almeida, L et al. The MUST guide to Paris – Implementation and expert evaluation of a multimodal tourist guide to Paris. *Proc. ISCA Tutorial and Research Workshop (ITRW) on Multi-Modal Dialogue in Mobile Environments (IDS 2002)*, Kloster Irsee, Germany, 49–51, 2002.
 - 18 Almeida, L et al. Implementing and evaluating a multimodal and multilingual tourist guide. *Proc. International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*, Copenhagen, Denmark, 2002.
 - 19 *GALAXY Communicator*. 2002, November 8 [online] – URL: <http://fofoca.mitre.org/>
 - 20 *Open Agent Architecture*. 2002, November 8 [online] – URL: <http://www.ai.sri.com/~oaa>
 - 21 *Adaptive Agent Architecture*. 2002, November 8 [online] – URL: <http://chef.cse.ogi.edu/AAA/>
 - 22 Wu, L, Oviatt, L, Cohen, P R. Multimodal Integration – A Statistical View. *IEEE Trans. on Multimedia*, 1 (4), 334–341, 1999.
 - 23 *Inovox*. 2002, November 8 [online] – URL: http://www.ptinovacao.pt/comunicacao/noticias/2002/jul2_inovox.html
 - 24 Knudsen, J E et al. *Tabulib 1.4 Reference Manual*. Kjeller, Telenor R&D, 2000. (R&D note N 36/2000)
 - 25 *Philips*. 2002, November 8 [online] – URL: <http://www.speech.philips.com/>
 - 26 *Microsoft Speech API 4.0*. 2002, November 8 [online] – URL: <http://research.microsoft.com/srg/docs/>
 - 27 *Code-it Software*. 2002, November 8 [online] – URL: http://www.code-it.com/tts_engines.htm
 - 28 *Telenor Talsmann*. 2002, November 8 [online] – URL: <http://www.telenor.no/fou/prosjekter/taletek/talsmann/>
 - 29 *XSLT*. 2002, November 8 [online] – URL: <http://www.w3.org/TR/xslt>
 - 30 *ActiveX*. 2002, November 8 [online] – URL: <http://www.microsoft.com/com/tech/ActiveX.asp>
 - 31 *Microsoft Jscript*. 2002, November 8 [online] – URL: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsoriJScript.asp>
 - 32 *PlatiNet*. 2002, November 8 [online] – URL: <http://www.platinet.com/>
 - 33 *SFR*. 2002, November 8 [online] – URL: <http://www.sfr.com/en/index.jsp>
 - 34 *NMS*. 2002, November 8 [online] – URL: <http://www.nmscommunications.com>
 - 35 *OnMobile*. 2002, November 8 [online] – URL: <http://www.onmobile.com/>
 - 36 *Kirusa*. 2002, November 8 [online] – URL: <http://www.kirusa.com/>
 - 37 *SpeechWorks*. 2002, November 8 [online] – URL: <http://www.speechworks.com/>
 - 38 *VoiceXML*. 2002, November 8 [online] – URL: <http://www.voicexml.org/>
 - 39 *W3C Multimodal Interaction Activity*. 2002, November 8 [online] – URL: <http://www.w3.org/2002/mmi>
 - 40 *W3C XHTML+Voice Profile*. 2002, November 8 [online] – URL: <http://www.w3.org/TR/xhtml+voice/>
 - 41 *SALT Forum*. 2002, November 8 [online] – URL: <http://www.saltforum.org/>
 - 42 *PVM home page*. 2002, November 8 [online] – URL: http://www.epm.ornl.gov/pvm/pvm_home.html
 - 43 *MPI home page*. 2002, November 8 [online] – URL: <http://www-unix.mcs.anl.gov/mpi/>
 - 44 *RPC-XML home page*. 2002, November 8 [online] – URL: <http://www.xmlrpc.com>
 - 45 SOAP protocol specifications. 2002, November 8 [online] – URL: <http://www.w3.org/2002/xp/Group/>

Multimodal Interaction – Will Users Tap and Speak Simultaneously?

JOHN RUGELBAK AND KARI HAMNES



John Rugelbak (55) received his *Siv.Ing. degree (MSc) in electrical engineering in 1973 and has been a Telenor R&D employee since 1974. He has been working in the field of subjective and objective user experiments since 1980. Since 1995 he has been a member of Telenor's Speech technology group, working mainly with speech dialogues and user experiments. For the last two years he has been working with multimodal systems. His main research interests are design and usability evaluation of speech and multimodal interfaces.*

john.rugelbak@telenor.com



Kari Hamnes (39) is an *HCI (Human-Computer Interaction) researcher in the Future Media Group at Telenor R&D. She obtained her MSc in Electrical Engineering and Computer Science from the Norwegian University of Science and Technology (NTNU, formerly NTH) in 1986. She has studied Human-Computer Interaction at University College London (1992–95), focusing on the use of usability guidelines in product development. Her research interests include multi-modal user interfaces, mobile user interfaces and the role of usability in product development processes.*

kari.hamnes@telenor.com

Well-designed multimodal interfaces can solve existing user interface problems, particularly for small handheld devices that do not allow mouse or keyboard input. For such devices, the combination of pen and speech input has proved to be efficient and effective, since the two modalities are complementary. With multimodal interaction it is easier to avoid and correct recognition errors, and dialogue completion times can be shortened.

The next generation of multimodal interfaces must not only offer increased functionality and efficiency for expert users but must also be user friendly, natural and intuitive for naïve users. But what is natural and intuitive interaction between humans and machines? When humans communicate with each other, we use co-ordinated speech, gestures, body language and facial expressions, and we combine different input senses such as vision and hearing. Communication between humans is by nature multimodal, and it is natural to use different modalities simultaneously and with low effort. Since this is a natural way for humans to communicate, it has been considered to be natural to communicate with machines in the same way.

However, this is not necessarily the case, and a main goal for the EURESCOM project MUST – “Multimodal and Multilingual Services for small Mobile Terminals” [1] has been to obtain knowledge about user behaviour with an application that supports simultaneous coordinated pen and speech interaction.

Simultaneous coordinated multimodal interaction is a term used by the World Wide Web Consortium [2] for the most advanced and powerful form of multimodal interaction, where all available input channels are active simultaneously, and their actions are interpreted in context. For a pen-speech enabled application this means that it is possible for the user to tap while he talks, and that the different modality actions are then interpreted together.

This paper reports the Norwegian part of an expert evaluation that was run prior to the MUST user studies.

The MUST Tourist Guide – A Sample Application Using Simultaneous Pen and Speech Input

In the MUST project Telenor cooperated with researchers from France Telecom, Portugal Telecom, Max Planck Institute and the University of Nijmegen. An important part of the project was to run experiments with the purpose of investi-

gating natural multimodal user interaction. We therefore needed:

- A platform that supported simultaneous co-ordinated multimodal input; and
- A test application/service where the user could complete tasks using different modalities one by one, or simultaneously.

Most experiments that have been run and reported previously have been based on Wizard of Oz platforms. In this project, it was decided to implement a working platform and a demonstrator/application where small experiments could be run with relatively low effort. The MUST PDA based platform is described in detail in [3a], [3b], [3c]. It was decided to implement an electronic map based tourist guide for Paris. The test application and user interface are described in the following.

Maps and Points Of Interest (POIs)

The tourist guide is organized as regional Paris maps (Figure 1), centred around different Points of Interest in Paris, such as Notre Dame, Hotel de Ville, the Eiffel Tower, Sacre Coeur, etc.

From an overview map of Paris (Figure 3) showing all available Points of Interest, the user can navigate to a regional map by tapping the POI or by saying the POI name. To move from one regional map to another, the user can either go via the overview map by tapping a button on the

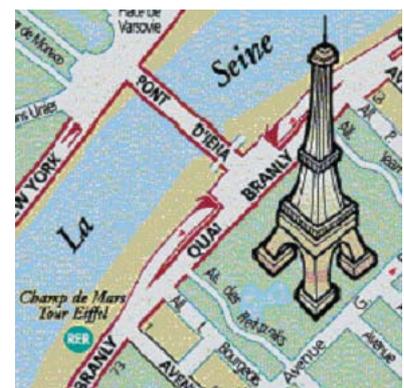


Figure 1 Regional map of Paris

toolbar, or he can use “speech shortcuts” and for example say “Show hotels near Notre Dame”.

Buttons

Figure 2 shows buttons that are present in the PDA’s tool bar. The first two can be used to select a Facility group, e.g. hotels or restaurants. Button number three is used to go back to previous map and the fourth to go to the overview map. The fifth button is used to end the present interaction, and the sixth to request help.

Facilities

On each regional map, the user can use voice or a Tool bar button to display different facilities. Since a main goal for the project was to make a demonstrator to run experiments, rather than to implement a service, the number of POIs, facilities, etc. was limited – but sufficient to run scenario-based experiments. For the first version, only hotels and restaurants were implemented. When voice is used to display facilities, it is possible to select subsets of each facility group, for example by saying “three star hotels” or “cheap Italian restaurants”.

Selecting Objects on the Map

POIs can be selected and made the topic of the dialogue by tapping the pen or by saying the name. The user does not have knowledge about any other objects, and these objects can therefore only be selected and made active by tapping the pen on the object.

Requesting Information About an Object

To request information about an object, the user must use voice. Address, telephone number, which type of food is served at a restaurant, opening hours or a detailed description of a POI are examples of information that can be requested.

“Tap while Talk” Functionality

An important functionality of the tourist guide is that the modalities voice and pen can be used one by one in a serial way, or simultaneously, which can be more efficient. If the user for example wants to select a hotel in the Notre Dame area and request the double room rate, he can either use modalities one by one and go through a four-step procedure:

- 1 Select the Notre Dame regional map
- 2 Display a group of facilities (hotels)
- 3 Select a hotel
- 4 Request info

or he can use pen and speech simultaneously and tap on an object while he talks:

- 1 “Show hotels here”
- 2 “Get double room rate for this”



Figure 2 Toolbar buttons

For this experiment we have defined “simultaneous” as “when pen is tapped within the time window one second before “speech detected” till one second after “end of speech”. The two actions are then integrated into one combined action and regarded as one dialogue turn.

Dialogue Strategy and System Output

The overall dialogue strategy is user controlled, in accordance with what is normal for graphic user interfaces. As a consequence of this, the speech recogniser must always be open for speech input. The system’s response to the user is mostly graphics (maps displaying POIs, hotels or restaurants) or text (requested information about objects on the map). Synthetic speech is used to give additional information “we found four such restaurants”, “we found no such hotels”, or to give the user error messages such as “I didn’t understand”.

A Study of Simultaneous Pen and Speech Interaction

Aim of Experiment

The aim of the experiment reported in this paper was twofold:

- To explore the “naturalness” of simultaneous pen and speech interaction; and
- To evaluate a sample application in order to improve its usability prior to subsequent user studies.

The first part of the aim was intended to help identify the main research questions for further study. The second part of the aim was intended to maximise the effect of a larger planned study with potential end-users (novices). By eliminating potential usability problems in the sample application, the study would be able to focus better on issues related to the pen and speech based interaction styles.



Figure 3 Overview map of Paris

| Goal | Action |
|---|---|
| 1 Check opening hours and entrance fee for Eiffel Tower | 1 Tap: on <i>Eiffel Tower</i> 2 Say: "what are the <i>opening hours</i> ?" 3 Say: "what is the <i>entrance fee</i> ?" |

Example 1 Example of pre-defined action sequence in Cognitive Walkthrough

This paper focuses mainly on the first part of this aim.

Subjects

In order to gain maximum effect of this study, we chose to use usability/user interface experts as subjects, as these experts would be able to offer well-founded comments with respect to both the naturalness of the interaction style, and the potential usability problems of the sample application.

The study included seven expert subjects, four males and three females. While all subjects had some or extensive experience with pen based interfaces, six were also familiar with speech interfaces. All subjects had been working several years within the field HCI/Usability; five subjects had some or extensive experience designing graphic interfaces.

Cognitive Walkthrough

The experiment was based on the Cognitive Walkthrough (CW) method. CW focuses on ease of learning by exploration and evaluates each step necessary to perform a task. The technique is based on a simplified 4-step model of learning by exploration [4]. Extensive practitioner's guides to cognitive walkthrough are provided in [5] and [6]. The technique itself will not be described in detail in this paper.

The experts performed the analysis stage of CW by walking through a set of predefined action sequences and recording problems related to interaction style and usability issues for each step of the sequence.

Experimental Procedure

The experimental procedure consisted of five steps:

- Introduction to experiment
- Exploratory phase
- Cognitive Walkthrough introduction
- Cognitive Walkthrough analysis
- Debrief interview

In the Exploratory phase, the subject was first asked to freely explore the prototype and comment on the interaction style or on apparent usability problems. In the second part of the exploration, the subject was asked to perform tasks of the type "Display the local maps for the Montmartre and Hotel de Ville", "Display hotels

near Notre Dame", or "Find Cuban restaurants near Notre Dame".

Having read a brief introduction to the Cognitive Walkthrough technique, the subject and the experimenter jointly performed a CW analysis on an example pre-defined action sequence in order to demonstrate how the technique works. The subject then performed a Cognitive Walkthrough analysis for three pre-defined action sequences, and identified problems in the design.

The semi-structured debrief interview focused on a number of pre-defined issues related to naturalness of interaction and usability of the MUST prototype.

Data Collection and Analysis

The Exploratory Phase and the subsequent initial comments, the Cognitive Walkthrough and the semi-structured debrief interview were recorded on audio and video.

The subjects talked aloud during the experiment to elaborate on problems, reasoning and possible design solutions. In addition, they recorded key words on the cognitive walkthrough form provided. A sample sequence is shown in Example 1.

During all sessions, the experimenters made notes of observations and comments from the subject.

The data analysis focused on three main issues, directly related to the two-fold aim of the study:

- Subjects' observations about the pen and speech interaction styles;
- Subjects' identification and reasoning about potential usability problems related to the interaction style and the specific MUST application;
- The experimenters' observation of the subjects' pen and speech interaction style.

The analysis was qualitative and relied on reviewing audio and video materials along with the subjects' and the experimenters' written comments and observations.

The audio/video was used for support in recording the problems and reasoning on the two main issues, and some of the contents was transcribed for exemplification of subjects' statements.

The videos were reviewed in detail for the Exploratory Phase (the tasks that include simultaneous speech and pen actions) with respect to timing issues.

Observations

Observations on Interaction Styles During Exploratory Phase

Although the users were told that pen and speech could be used simultaneously – and that this was our focus for the experiment – three of the seven subjects never used pen and speech simultaneously. They used pen and speech as clearly separated actions, and the most typical behaviour was to use pen to select a facility group, a single facility or a POI, and then to use speech to request information.

The typical behaviour of the remaining four was to first use modalities one by one to explore the system for a while, and then try to use both simultaneously.

Observations on Timing During Exploratory Phase

Figure 4 illustrates the timing between pen and speech during the explorative phase. The approximate timing of when the pen is used is indicated with an arrow (↓).

From Figure 4, we see that

- The users used deictic words like “here”, “there” or “this” when they combined pen and speech. 14 out of 16 utterances contained deictic terms and for 12 of these 14, the deictic term was the last or second last word of the utterance.
- The users tended to tap near the end of the sentence (13 out of 16 utterances), but the timing seemed to be even stronger correlated to the use of deictic terms than to the sentence end. Since the deictic term in most cases occurred close to the sentence end, the users almost always tapped close to a deictic expression as well as to the sentence end. However, when the deictic term occurred early in an utterance, the user also tapped early.

Observations on Timing During Cognitive Walkthrough

For all dialogue turns that include both pen and speech, the timing between pen and speech is shown in Figure 5. Note that all sentences in this figure are predefined scripts that the users should follow, but it is up to each user to decide when to tap.

We see that there are some individual differences. It seems that users 1 and 5 prefer to tap a little earlier than the others. (Expert 1 also showed a typical tap-then-talk behaviour in the explorative phase.) However, all experts tend to tap at the end or shortly after the sentence. The largest variance is found for Task 3.1.2. This task involves the only sentence that does not

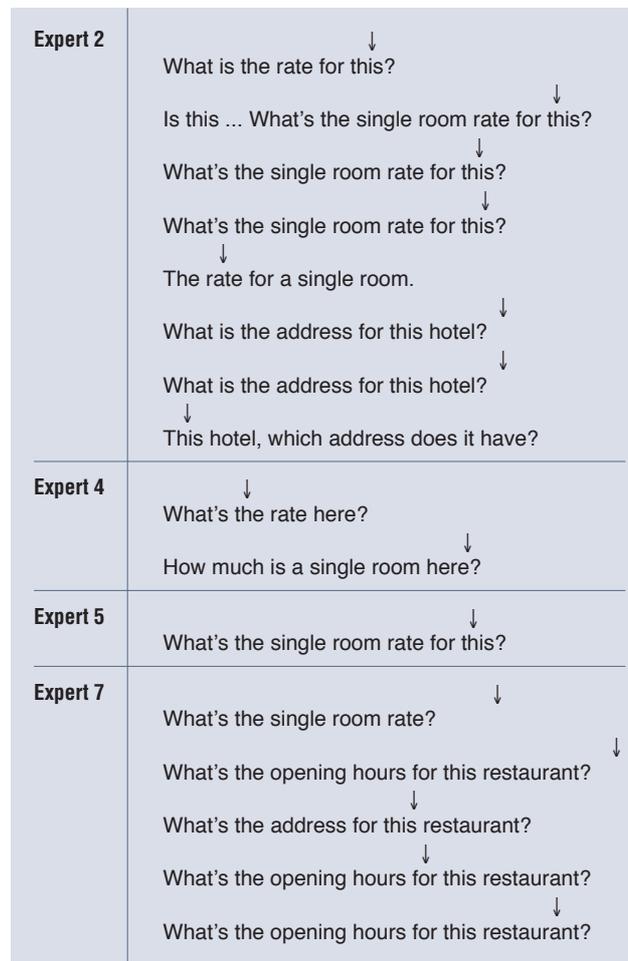


Figure 4
Timing during
Exploratory
Phase

contain deictic words. All other sentences have deictic words at the end. Since the users were scripted to use words like here and there, we have reason to believe that through this we influenced the users' interaction style.

Results from Cognitive Walkthrough Analysis

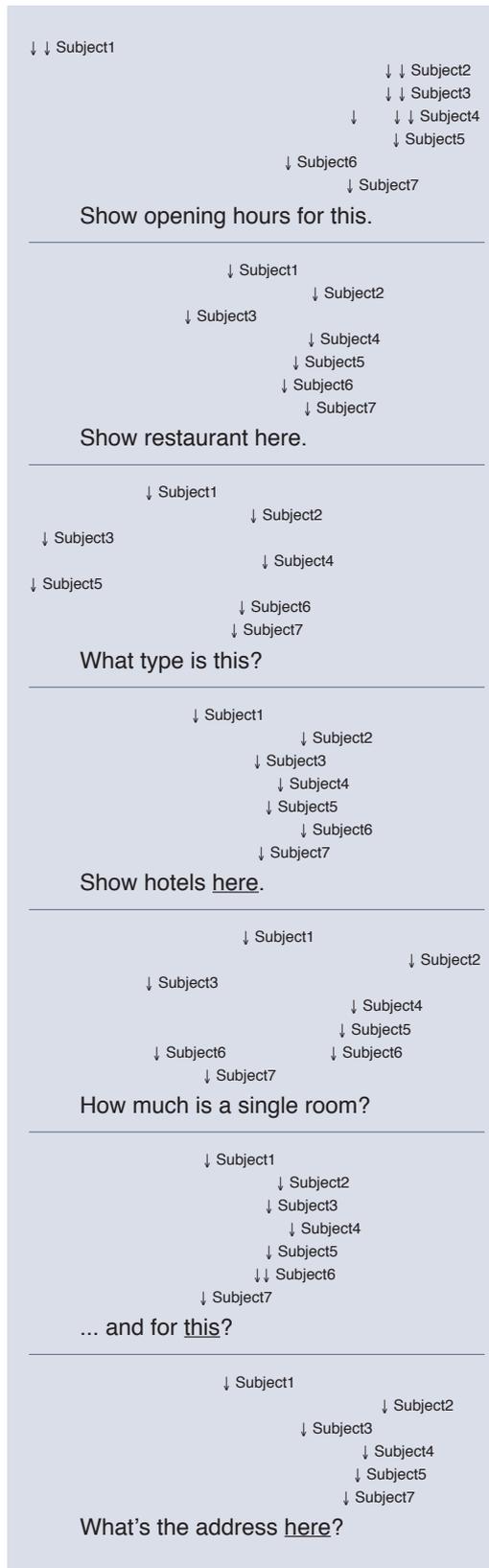
The problems identified by the experts in the Cognitive Walkthrough were classified as belonging to one of 19 main design issues. These issues were further categorized according to the main aims of the evaluation, namely whether they related to the pen and speech interaction styles or to the usability issues specific to the MUST application. This paper will only discuss the six interaction style issues

- Domain knowledge
- Prompting
- Training/Instruction
- “Tap” to select
- Timing
- Speech as shortcut

Domain Knowledge

Several experts commented that the interaction style (particularly the speech part) would be more intuitive and work better in a domain in which the user is familiar. The MUST Tourist

Figure 5 Timing during Cognitive Walkthrough



Guide requires domain knowledge in the form of detailed knowledge of Paris and buildings in Paris, as well as knowledge about what type of information it is possible to get. Knowledge of the domain would also help the user by forming expectations with respect to vocabulary. Yellow Pages (YP) is an example of a “domain” which most users know. They know that YP contains various classes of professions, and contact infor-

mation for the professionals or businesses, and the users have a fairly good idea of the vocabulary they can use.

Prompting

Several experts commented that prompting could be one strategy for encouraging users to explore the multimodal interaction style. The users could be given hints about the available functionality during the dialogue, for example that it is possible to use speech- or combined speech and pen shortcuts.

Training/Instruction

All experts agreed that without some initial training and instruction, users would probably not use a multimodal interaction style. Indeed, initial training/instructions is a requirement to even understand that the MUST Tourist Guide is multimodal. The screen does not indicate that it is possible to use speech. It is not intuitive that it is possible to use speech at all, and in particular to use pen and speech simultaneously, or to use shortcuts related to objects that are not visible.

“Tap” to select

Several experts commented that PDA users would be more inclined to select objects (e.g. POIs) by tapping, as opposed to selecting objects using speech, due to previous learning. One expert commented that she would probably tap, tap, tap – until there are no more choices, and then try to speak.

Another comment was that when one has a limited domain, and does not exactly know which alternatives are available, a PC or PDA user is used to tapping or using the mouse again and again, to narrow the “search space”.

Timing

The experts commented on the timing issue of simultaneous coordinated input. In general, they appreciated the functionality and indeed felt that it was quite natural after having used it for a little while. However, they felt that users would be unsure about when they would have to tap in relation to what they said. Many of the experts said that it felt more natural to tap towards the end of the sentence. Several experts said they would feel it as an unwanted restriction, if they had to tap exactly during speech. A user-friendly system should therefore be flexible regarding when the user is allowed to tap: The system should allow the user to tap during speech, as well as shortly before or after.

Speech as Shortcut

It was mentioned that PDA users would be more likely to tap, in general, but that speech/multimodal interaction could have a potential as shortcuts to specific data. It is however not in-

tuitive that one can request information about objects that are not visible.

Discussion

Naturalness of Simultaneous Pen and Speech Interaction

Since only seven experts participated in this evaluation, results should be interpreted with due caution. The most noteworthy observations will be discussed here.

During the Exploratory Phase of the evaluation, most experts started to use the two input modalities one by one, and some of them never tried to use them simultaneously. After a while, four of the seven experts started to use pen and speech simultaneously.

Timing between speech and pointing has been studied in other experiments, e.g. [7] and [8]. In the expert evaluation we observed that the experts typically tapped at the end or shortly after the utterance. This was especially the case when the utterance ended with deictic expressions like 'here' or 'there'. If no deictic expressions were present, tapping often occurred somewhat earlier. Timing relations between speech and pointing will be investigated in more detail in the user evaluation experiment that is now being designed.

The results from the Exploratory Phase indicate that frequent PC and PDA users are so accustomed to use a single modality (pen or mouse) to select objects or navigate through menus to narrow down the search space, that even if they are told that it is possible to use speech and pen simultaneously, they will have to go through a learning process to get accustomed to the new simultaneous coordinated multimodal interaction style. But once they have discovered and experienced it, the learning curve appears to be quite steep.

It was not intuitive and obvious that the interface was multimodal, and in particular that the two modalities could be used simultaneously. This indicates that for the naïve user evaluation we should pay much attention to the introduction phase where we explain the service and the interface to the user.

During the expert evaluation many usability issues were revealed. They can be divided into interaction style issues and issues that are specific for the MUST tourist guide. The MUST guide specific issues were mainly related to buttons, feedback, prompts, the way selected objects were highlighted, and the location of the POIs on the screen. Most of the problems can be solved rather easily. The comments from the experts gave helpful advice to improve the graphic interface and button-design for the sec-

ond version of the demonstrator that will be used for the user evaluation experiments.

Almost all experts agreed that without some initial training and instruction, the users would probably not intuitively use a simultaneous multimodal interaction style. They also believed that the users would probably be able to use such an interaction style with small cognitive effort, once they are aware of the systems capabilities. This is also supported by our observations of the experts' behaviour during the Exploratory Phase.

With the present lack of multimodal applications for the general public, there is a need to introduce the capabilities of simultaneous coordinated interaction explicitly before customers start using the new products. According to the experts a short video or animation would be suitable for this purpose. The introduction that is given to the users before they start to use the tourist guide will be the main parameter in this experiment.

In the introduction to the explorative phase, the experts were explicitly instructed to use the two modalities both one by one and simultaneously. Still only four experts used simultaneous interaction, and only 16 out of approximately 250 to 300 dialogue turns were "simultaneous" (contained both pen and speech). The far most typical interaction style was to use modalities one by one. There are several possible explanations for this, such as the fact that users are accustomed to operating graphic interfaces in a serial manner. Another possible explanation is the cognitive load associated with pointing and speaking simultaneously. During inter-human dialogue, speech and pointing actions are occurring simultaneously, obviously without effort. This also includes the use of available aids such as pencils and pointers. In [9] these pointing actions are denoted "Natural Pointing". Simultaneous multimodal systems have made it possible to simulate gestures and pointing found in inter-human communication. However, for these systems, the user must also touch a small object on a screen ("Tactile Pointing"). If the user speaks and uses Tactile Pointing simultaneously, it is likely that there will be a resource competition between talking and pointing, and that this cognitive load is sufficiently large to influence the user's choice of interaction style (use modalities simultaneously or one by one).

Conclusions and Future Work

The main goal for this experiment has been to identify research issues to be studied further in a planned user experiment within the MUST project. Seven experts in the fields HCI and Usability participated in an experiment supporting simultaneous pen and speech input.

The main conclusions and topics for further study were that:

- This is a new way of interacting with machines, and the users will need an introduction to understand or be aware of this new functionality (that it is possible to both tap and speak, and particularly that it is possible to do both simultaneously). An animated instruction (video) showing “how to do it” may be more effective than text.
- It is not intuitive or natural for new users to tap and speak simultaneously. They are used to operating PCs, PDAs etc. in a sequential way, and the typical behaviour will probably be to tap, tap, tap etc. and then speak. Even when they are aware of the simultaneous functionality, they may choose to use the interface sequentially, because of the larger cognitive load. But users seem to “learn” quickly and the cognitive load will be smaller when users become expert users. Since speech centric multimodal interfaces are new, there is little research data on the mental effort the user spends in processing multimodal input, and we see this as an interesting research question.
- When users tap and speak simultaneously, and the utterance contains a deictic word, there seems to be a strong timing relation between pen and the deictic word. If deictic words are actively used in the introduction and system prompts, it may be possible to influence the users’ pen timing and interaction style, since the users will probably mimic words used by the system.

The experts agreed that multimodal pen and speech systems have a great potential, and that users can and will use such interfaces. To what extent users will tap and speak simultaneously will depend on at least three issues:

- 1 Whether they will continue to use the serial interaction style they are used to when they operate graphical interfaces (PCs, PDAs).
- 2 Whether the cognitive load associated with using two modalities simultaneously is sufficiently low. If not, the users may prefer to use modalities one by one.
- 3 The application and how much there is to gain by using pen and speech simultaneously. If a user finds simultaneous interaction style more efficient – or maybe a must, he probably can and will use pen and speech simultaneously, even if he normally prefers to use modalities one by one.

References

- 1 *EURESCOM Project p1104*. 2003, June 18 [online] – URL: <http://www.eurescom.de/public/projects/p1100-series/p1104>
- 2 *Multimodal Requirements for Voice Markup Languages*. W3C Working Draft, 10 July 2000. 2003, June 18 [online] – URL: <http://www.w3.org/TR/multimodal-reqs>
- 3a Almeida, L et al. The MUST guide to Paris : Implementation and expert evaluation of a multimodal tourist guide to Paris. *Proc. ISCA tutorial and research workshop on Multi-modal dialogue in Mobile environments (IDS2002)*, Kloster Irsee, Germany, 2002.
- 3b Almeida, L et al. Implementing and evaluating a multimodal tourist guide. *Proc. International CLASS workshop on natural, intelligent and effective interaction in Multimodal dialog system*, Copenhagen, Denmark, 28–29 June 2002.
- 3c Almeida, L et al. User friendly multimodal services, a MUST for UMTS. *EURESCOM Summit*, Heidelberg, 21–24 October 2002.
- 4 Polson, P G, Lewis, C. Theory-based design for easily learned interfaces. *Human-Computer Interaction*, 5 (2 & 3), 191–220, 1990.
- 5 Wharton, C et al. The cognitive walkthrough method: A practitioner’s guide. In: Nielsen, J, Mack, R L (eds.). *Usability inspection methods*. New York, NY, John Wiley, 1994.
- 6 Lewis, C, Wharton, C. Cognitive walkthroughs. In: Helander, M, Landauer, T K, Prabhu, P (eds.). *Handbook of Human-Computer Interaction*. New York, Elsevier, 1997.
- 7 Martin, J-C, Braffort, A, Gherbi, R. 2000. Measurement of cooperations between pointing gestures and constrained speech during human-computer interaction. *3rd International Conference on Methods and Techniques in Behavioral Research “Measuring Behavior 2000”*, Nijmegen, The Netherlands, 15–18 August 2000.
- 8 Kehler, A et al. 1998. On Representing Salience and Reference in Multimodal Human-Computer Interaction. *Proceedings of the AAAI’98 workshop on Representations for Multi-modal Human-Computer Interaction*, Madison, Wisconsin, 26–27 July 1998.
- 9 Schmauks, D. Natural and simulated pointing. *Proceedings of the 3rd European ACL*, Copenhagen, Denmark, 1987, 179–185.

A Norwegian Spoken Dialogue System for Bus Travel Information

– Alternative dialogue structures and evaluation of a system driven version

MAGNE HALLSTEIN JOHNSEN, TORE AMBLE AND ERIK HARBORG



Magne Hallstein Johnsen (52) received his *Siv.Ing. (MSc)* and *Dr.Ing. (PhD)* degrees in Electrical Engineering from the Norwegian Institute of Technology (NTH) in 1974 and 1981, respectively. He has been assistant professor at NTH, scientist at SINTEF DELAB, and head of the Signal Processing Group at Chr. Michelsen Institute (CMI), Bergen, where his main activity was helium speech enhancement from divers. He joined the Faculty of Electrical Engineering and Computer Science in 1985 as associate professor at the Dept of Telecommunications. His main fields of interest are digital signal processing, pattern recognition, and automatic speech recognition. Dr. Johnsen is a member of IEEE, NOBIM and NORSIG. mhj@tele.ntnu.no



Tore Amble (58) is associate professor at the Department of Computer and Information Science, The Norwegian University of Science and Technology (NTNU), where he does teaching and research in the areas Knowledge Based Systems, and Computational Logic and Natural Language Interfaces. Among his merits can be mentioned a textbook "Logic Programming and Knowledge Engineering" (Addison-Wesley, 1989). Since 1989, he has been engaged in projects aimed at constructing operational natural language interfaces to information systems.

amble@idi.ntnu.no

This paper describes the development and testing of a pilot spoken dialogue system for bus travel information in the city of Trondheim, Norway. The system driven dialogue was designed on the basis of analysed recordings from human-human operator dialogues, Wizard-of-Oz (WoZ) dialogues, and a text-based question-answering system (BusTUC) for the web. The dialogue system employs a flexible speech recogniser and an utterance concatenation procedure for speech output. The dialogue manager was integrated with a modified version of BusTUC, called TABUSS.

Even though the system was intended for research only, it was accessible through a public phone number from October 1999 until end of 2001. During this period all dialogues were recorded. From the first year recordings, approximately 350 dialogues were selected and compared to 120 dialogues from the WoZ recordings with respect to recognition error rate and turn error rate.

The experiments showed that the turn error rate was more than twice as large for the real dialogues as for the WoZ calls, i.e. 13.3 % versus 5.7 %. Thus, the WoZ results did not give a reliable estimate for the true performance. Our system driven experiments indicate that the current flexible speech recogniser should be further optimised. This was verified by an initial experiment with a mixed initiative version of the system (BUSTER).

1 Introduction

Spoken dialogue information systems over the telephone line have enjoyed growing commercial interest over the last few years and a large number of such systems have been developed and tested. The systems vary a lot in complexity, not only due to the variation in tasks and design techniques, but also because of the difference in targeted user friendliness. Recently, standard methods for evaluation of spoken dialogue systems have been suggested [1].

The main components in a spoken dialogue system are the speech recogniser and text-to-speech modules, the natural language processing (NLP) unit and the dialogue manager. The dialogue manager keeps track of the dialogue state and history, accesses the database, and forms a textual output prompt for the text-to-speech module. The NLP unit performs the semantic interpretation of the output from the speech recogniser. This unit is interfaced with the dialogue manager, and can employ knowledge about the current dialogue state and the dialogue history.

The dialogue style has a large impact on the system complexity. The ultimate goal is to develop spoken dialogue systems with a natural and flexible dialogue. This implies that both the system and the user should be able to influence the dialogue flow, and is often termed a *mixed initiative* system [2]. The user is allowed to take control of the dialogue, i.e. give subjectively relevant information in a possibly over-informative manner, correct, negate or verify the system hypotheses. In practice, this normally means that user speech barge-in must be accepted.

In a *system driven* spoken dialogue system, the users themselves cannot take much initiative, but are restricted to answering the questions posed by the system. Some minimum user control is however always required, e.g. in order to ask for help or to negate the last system hypothesis. For a given task, a user will normally prefer the mixed initiative dialogue over the system driven one, but this obviously involves a large increase in system complexity. Also, even the mixed initiative system requires the option of using a system driven dialogue as a fallback strategy, typically in situations where the speech recogniser breaks down due to a poor signal.

This paper describes the first version of a spoken dialogue system for Norwegian. The pilot system TABUSS was implemented for the domain of bus travel information for the city of Trondheim in Norway. This task was chosen for several reasons. First of all, the dialogue has a manageable complexity. We also had access to the bus company's databases, and a text-based NLP question-answering system, BusTUC [3], already existed. Our primary goal was to develop a demonstrator that could form the basis for further research, and a system driven approach was chosen. Further, initial work was done in order to extend the system towards a mixed initiative spoken dialogue version in the same domain.

2 The Preliminary Data Analysis

In order to get a coarse impression of the way naive users communicate with an operator, more than 100 real human-human dialogues were recorded and annotated. These recordings showed that



Erik Harborg received his Dr. Ing. degree in Electrical Engineering from the Norwegian University of Science and Technology (NTNU) in 1990, and has been employed by SINTEF Telecom and Informatics since 1989. His background is within various areas of speech technology, like speech compression, speech recognition and spoken dialogue systems, and he has been project manager of many national projects within this area. Dr. Harborg has authored/co-authored more than 20 publications within the speech technology area in international, refereed journals and conferences. He is a member of NIF, NORSIG and IEEE.

erik.harborg@sintef.no

- In order to *simulate* real dialogues accurately, a very complex mixed initiative dialogue structure would be required;
- In contradiction, a straightforward system driven approach would fulfil *most of the user goals* (given motivated users).

The web-based question-answering system BusTUC [3] has been operational for several years, and a significant amount of inquiry text data is thus available. BusTUC requests the user to formulate a complete inquiry in a single, preferably grammatically correct sentence, and thus there is no real dialogue involved. The logs from BusTUC and the annotated real dialogues were analysed in order to extract typical user goals and formulations including a corresponding vocabulary. Further, these goals were used to design a dialogue structure that should help the user to reach these goals.

On the basis of the above logs, the first version of a system driven dialogue structure was designed and implemented for use in a WoZ set-up. The wizard performed a controllable wizard-driven (human-human) dialogue and collected query items (slot filling) until a question could be forwarded to BusTUC.

The WoZ dialogue system was tested for a period of eight weeks. During this time, the dialogue structure was continuously improved. The resulting structure was then used to record 150 WoZ dialogues. These recordings were then annotated according to the SpeechDat standard [4].

3 The System Driven Dialogue Structure

A slightly simplified version of the final dialogue structure is shown in Figure 1. The dialogue manager has a total of ten logical states. Each of these states has a distinct dialogue manager stage and a small set of user prompts. The ten states are mapped to only five different 'recognition' states, corresponding to the following five sub-tasks:

- *Yes/no*: In five different dialogue manager states, the user is prompted to answer yes or no.
- *Bus stops*: In two dialogue manager states ('Place of departure' and 'Place of arrival'), the user is prompted to give the name of the bus stop. The vocabulary in these states includes nearly 600 names.
- *Day of week*: The user is prompted to say the day of travel. The vocabulary contains the seven days of the week, in addition to expressions like 'today' and 'tomorrow'.

- *Time information*: The user is prompted to give the desired travel time. This is actually the most complex network (though not the largest vocabulary), both due to pronunciation and syntactic variations which are experienced in Norwegian [5].
- *Departure/arrival*: The user is prompted to answer whether the specified time relates to departure or arrival.

In addition to the state specific vocabularies, all states included a common set of words for control, negation and help.

The dialogue system is implemented on a Linux-based PC. A software library supplied by TeleNor R&D, TabuLib, has been used for handling of the ISDN interface, the speech I/O module interface and for interface to the HAPI recogniser [6].

4 The Speech I/O Part

4.1 The Speech Recogniser

A flexible speech recogniser was designed according to the procedure developed in COST Action 249, *Continuous Speech Recognition over the Telephone* [7]. The Norwegian SpeechDat database [8] was used for training. At the time of the experiments this database was restricted to 1000 speakers, which was recorded over ISDN-based fixed lines. The resulting acoustic models consisted of a context dependent phone set with a relatively strong degree of state tying. The HMM state output observations were modelled by 8-component Gaussian mixture models. In addition to the triphones, models for the SpeechDat defined noise labels (man-made and background) were trained. Finally, to cope with Out-Of-Vocabulary (OOV) words, a lexical-based filler model was designed from a set of monophones, according to a simplified version of the procedure presented in [9].

The overall vocabulary consisted of approximately 700 words, of which nearly 600 represented different bus stop names. Less than 100 of the vocabulary words existed in the SpeechDat dictionary, and thus an initial lexicon for the remaining words was designed by an experienced phonetician. This lexicon included more than 1500 entries, i.e. an average of more than two pronunciations per word. The lexicon was pruned to 800 entries (1.1 pronunciations per word) in a semi-automatic manner. A part of this procedure involved forced alignment recognition on a subset of 30 dialogues from the WoZ recordings. With this new lexicon, the word error rate was significantly reduced for the remaining 120 WoZ recordings.

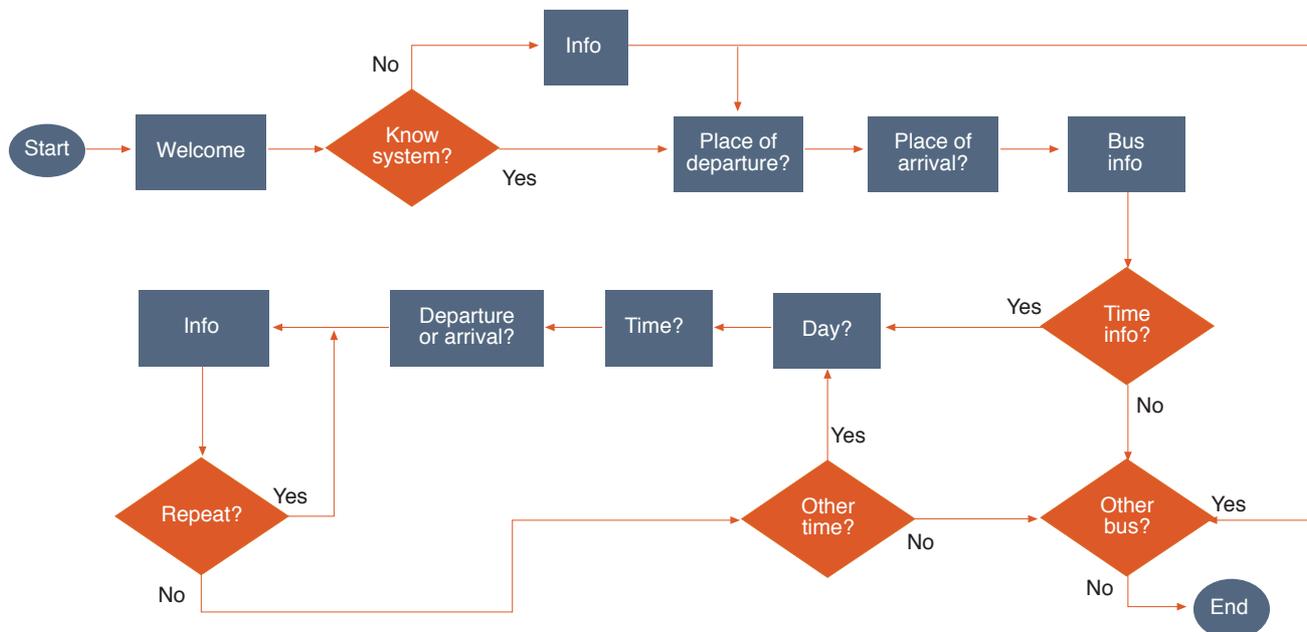


Figure 1 The system driven dialogue structure

Each of the five recognition modules were defined by a task specific finite state word network and corresponding vocabularies.

4.2 The Speech Output

Ideally, dialogue system prompts should be transformed into speech by a text-to-speech system (TTS-synthesis). The use of TTS makes it possible to use a variety of system text prompts. Further, to modify the system messages one normally only has to modify the text. However, the quality of synthesised speech is still significantly poorer than human speech, thus concatenated speech should be evaluated whenever possible.

In a system driven dialogue like TABOR, the number of possible spoken responses from the system is limited. Thus, it is feasible to utilise concatenation of pre-recorded utterances for the generation of speech prompts. For this purpose, a total of approximately 2000 sentences covering all the variables and fixed parts of the prompts were read by a single person. This included versions for both a flat and a falling intonation. The recordings were then manually edited prior to a simple gain normalisation. The procedure resulted in speech prompts which were found both pleasant and fully understandable.

5 The NLP Module and the Dialogue Manager

5.1 The BusTUC System

The NLP module for the text based question-answering system, BusTUC, is based on a complex set of rules and is implemented in Prolog. It is versatile with respect to understanding and answering a variety of alternative formulations requesting the same kind of information. This question-answering system consists of three

modules. The bottom module, BusLOG, includes the bus route database, the bus-stop list of names, and a route analyser/planner, which finds the shortest/best route between two given bus stops. The second module is a general text understanding module (Text Understanding Computer – TUC) which performs a rule-based grammatical and semantic parsing. The third and main module integrates TUC and BusLOG and tailors the system to process a complete inquiry in a single sentence. In fact, this is the part which is called BusTUC, as it is the part specifically made for the question-answering mode. Thus one could call this module a degenerate dialogue manager, i.e. with only one turn. BusTUC will thus perform a full sentence analysis. However there is no memory as there is no dialogue; i.e. every question concerns a new, independent inquiry. This can be regarded as an extreme variant of a so-called ‘User-initiative’ system. Thus, BusTUC will typically understand and respond to sentences like:

- I would like to travel from Ila to Saupstad in about one hour from now.
- Which bus(es) stop(s) at Nidar?

In the first case the system will inform about all bus numbers which pass the ‘from’ place until one hour after the time of the inquiry. In the second case the system will (default) anticipate that the user asks for the first bus passing by, however, it will give times and bus numbers for both directions.

5.2 The TABUSS System

The BUSLOG module from the above question-answering system was chosen as the basis for the dialogue manager in the speech-based system

called TABOR. As the dialogue now is system driven, the dialogue manager TABUSS, was designed to have a deterministic finite state structure, where only one semantic entity was expected/accepted for each turn (see Figure 1). Further, this entity was restricted to be either informative (bus stop, time, etc.) or a correction. An example of a possible dialogue is given below (note that the semantic entities are underscored):

- S: Welcome to the TABOR system. Do you know how the system works?
- U: Yes.
- S: Where do you want to travel from?
- U: (From) Blakli.
- S: You want to travel from Bakli. Where do want to travel to?
- U: No, that is not correct.
- S: Where do you want to travel from?
- U: (From) Blakli.
- S: You want to travel from Blakli. Where do want to travel to?
- U: I'd like to go to Stavset.
- S: You want to travel from Blakli to Stavset. Please wait. ... Bus number 12 passes Blakli on the way to Stavset. Do you want information about time?
- U: Yes.
- - - -

As seen from the above example, the inputs (outputs from the recognisers) have a much simpler form than for the BusTUC case. As a consequence, the understanding part, TUC, was not needed. TABUSS thus consists of BUSLOG and the main module. This main module (i.e. the genuine TABUSS) does the following operations:

- Parses the output from the recogniser to extract the semantic entity;
- Accumulates semantic entities;
- Keeps track of the dialogue state and predicts the next state;
- Sends requests to BusLOG;

- Composes an XML-tagged prompt based on returned information from BusLOG;
- Returns the XML prompt to the speech generating part.

TABUSS communicates with the speech front-end by asynchronous handshaking. The front-end is implemented as a Perl script, which calls TabuLib modules like the telephone interface, the recogniser and the speech prompt generator. Further, a simple dialogue manager is included in the front-end which

- Keeps control of the dialogue states in order to activate the corresponding state-dependent recogniser;
- Exchanges information between the different internal modules and to/from TABUSS;
- Analyses and decomposes the XML-prompts from TABUSS into fixed and variable parts for concatenative utterance generation.

5.3 The BUSTER System

BUSTER is a text-based and mixed initiative version of the inquiry system BusTUC. It was designed as a first step towards a corresponding mixed initiative version of TABUSS, i.e. a speech based system. As a special case, BUSTER accepts both BUSTUC type of complete text inquiries (single turn) and TABUSS type of single entity turns. A further goal was to be able to use BUSTER as a functional starting point for a speech based version. Thus the system is made robust with respect to inputs which reflects recognition errors in a corresponding speech based system. To accomplish this, BUSTER gracefully degrades the dialogue towards the TABUSS mode.

As BUSTER includes BusTUC and thus allows inputs with more than one semantic entity, both BusLOG and TUC were needed. The main module (i.e. the real BUSTER module) plays the role of a mixed initiative dialogue manager. This of course includes the tasks listed for the TABUSS module in section 5.2. However, in order to allow mixed-initiative turn-taking, a complex dialogue structure (relative to TABUSS) was implemented, based on a dialogue grammar. This grammar must be able to resolve problems like global anaphoric references and confirmation/denial combined with new information.

In order to increase the quality of service, system relaxation and acceptance of multiple sentences should ideally be included in the system. However, this is presently not implemented in BUSTER.

The following examples are meant to demonstrate system relaxation, multiple sentences, anaphora resolution and confirmation/denial:

- - - -
- U: When does the first bus after 18:00 leave Rotvoll towards the City Centre?
- S: No buses leave Rotvoll after 18:00. The last bus leaves at 17:40.
- - - -
- U: I have to go to from Dragvoll to the bus station. What bus must I take to be there before 12:00?
- - - -
- U: I would like to take the next bus from Munkvoll to the City Centre.
- S: Bus number 5 passes Munkvoll at 17:53.
- U: And when does the next bus pass Munkvoll?
- S: The next bus passes Munkvoll at 18:23.
- U: When does the first bus you mentioned pass Ila?
- - - -
- U: I would like to go to City Syd from Lundåsen around 13:00.
- S: Bus number 18 passes Lundåsen at 13:08. Do you want some more information?
- U: Yes, before 13:00
- S: Bus number 18 passes Lundåsen at 12:38. Do you want some more times?
- U: No, but the first bus to the City Centre.
- - - -

The use of a dialogue grammar as the basis for BUSTER has resulted in a reasonable generic dialogue manager. Most of the rules are thus also applicable to route planning for other domains.

BUSTER has been implemented as a Java client communicating with a server over the Internet.

6 Evaluation of the System

A preliminary evaluation of TABOR has been performed on the basis of the logged dialogue recordings. At the time of the experiments, there still did not exist any agreed upon standard for evaluation of dialogue systems. However, serious candidates, like the PARADISE scheme [1] had been proposed. In the evaluation metric for most methods, a subjective overall score from the user is normally included. However, due to mainly naïve users over the phone, we did not implement this.

6.1 The Evaluation Procedure of TABOR

Our main goals was to investigate:

- Which measures will give us an indication of the real demonstrator performance?
- Can the WoZ database be used for prediction of this real system performance?
- What are the main differences between the WoZ dialogues and the real system dialogues?

The turn error rate is believed to give a good indication of the quality of the recogniser performance for the TABOR system. The user success rate was harder to define and thus to estimate, as many users had more than one goal but did not get answers to all of them. This was mainly due to the use of out-of-vocabulary (OOV) names or out-of-domain (OOD) questions. One important reason that OOV names occurred was that commonly used aliases for the correct/formal name of bus stops were incorporated only to a limited degree. Thus, it is fair to claim that this version of TABOR in some cases requires local knowledge by the user in order to be able to pose successful queries to the system. In addition, many users violated the dialogue structure, usually by supplying information in an over-informative manner. The number of illegal turns should thus give a good indication of the quality of the dialogue structure. The number of turns per goal is also an indicator of the effectiveness of the dialogue structure. Both these parameters have been investigated.

For comparison, the same evaluation was performed for the WoZ recordings. The WoZ informants were given written scenarios, thus the relative amount of OOD and OOV utterances was modest. Further, the human Wizard responded both to over-informative responses (by extracting the relevant information) and to OOD queries (“I am sorry, but I cannot answer that”). Thus, the WoZ results can be regarded as a performance limit for the TABOR system. A turn error rate and a turn illegal rate close to the WoZ

results should thus indicate a system with a satisfactory recogniser performance and dialogue structure.

6.2 The TABOR Logs

The TABOR system is primarily intended for research issues, and the system lacks some important qualities which one would have to incorporate before offering the system as a commercial service. Despite this, it was decided to offer the current system to naïve users through a public telephone number. The main intention was to get a first impression of the popularity and usefulness of such systems.

Even though the demonstrator has not been announced to a large degree, a surprisingly large number of calls have been logged. Through the first period of approximately six months, more than 800 calls were received. However, more than half of these were discarded from further use due to one or more of the following reasons:

- The call was not a serious attempt to gain information. The typical scenario in this class is a call made from a party.
- The dialogue restrictions were deliberately violated.
- The dialogue restrictions were violated in an over-informative manner.
- The questions were mainly of the OOD-type.
- The user addressed alternately the system and people in the same room, but talked into the handset microphone all the time.

Obviously, our experiments and results will be strongly dependent upon the selection of calls (and discarding of illegal calls). We ended up with approximately 350 calls for further use. To a large extent, the users in these calls tried to follow the system dialogue structure. However, still most of the users cannot be characterised as cooperative. Thus we believe that these calls are representative of naïve but serious users of the spoken dialogue system.

Table 1 Evaluation of WoZ and real dialogues

| Number of: | WoZ | Real |
|--------------------|------------|--------------|
| dialogues | 120 | 350 |
| turns | 1412 | 3019 |
| turns per dialogue | 11.8 | 8.6 |
| illegal turns | 47 (3.3 %) | 321 (10.7 %) |
| turn errors | 80 (5.7 %) | 402 (13.3 %) |

7 Experiments and Results

In this section, we report the experiments performed with the two data sets we have available, i.e. the recordings from the WoZ callers and the real users. In both cases, the dialogues were annotated according to the SpeechDat standard and compared to the output from the speech recogniser. Turns with truncated and/or unintelligible speech were removed from the test set. Turns including OOV bus stops and/or OOD requests were defined as illegal turns. Table 1 summarises the results.

The results show a large difference between the WoZ and real dialogues with respect to the number of illegal turns. This is not surprising, as the WoZ callers were given written instructions, which described the scenario in some detail. It also seems like a large proportion of the real users did not have the proper knowledge about the system limitations. Despite this, only 10 % asked for directions of use. We hope this can be improved by a change in the welcome prompt. Although naïve users dominated also among the WoZ callers, they probably benefited from the implemented WoZ strategy in learning the system limitations. For instance, giving an intelligent answer like “Sorry, I cannot answer this kind of questions” to the OOD phrases is likely to improve the caller’s understanding of how the system works.

The turn error rate is defined as the relative number of turns for which the recogniser makes an error with respect to the semantic entity of that turn. Thus the errors result in a semantic misunderstanding, i.e. wrong place, time, day etc. The turn error rate is more than twice as large for the real dialogues compared to the WoZ dialogues. The performance difference of the speech recogniser for the two data sets was most evident for the largest word network, i.e. for recognition of the bus stops. The overall rather poor performance achieved in these experiments confirms that the speech recogniser needs further optimisation with respect to spontaneous speech. The difference in turn error rate can probably be related to several circumstances. Three obvious explanations were found by analysing the recordings:

- All the WoZ dialogues were recorded over a fixed ISDN-line, while a substantial part of the naïve users applied a mobile phone.
- The WoZ callers were more motivated with respect to using a distinct pronunciation.
- The high number of illegal turns (see below) had a negative impact on the real user dialogues, not only with respect to success rate but also with respect to the speaking style (irritation, etc.).

We also did some initial experiments with a mixed initiative version using the speech front-end together with BUSTER. As a first step the dialogue was restricted to only accepting one or both of the departure/arrival names, however embedded in a freely pronounced utterance. If the user starts by giving only one of the places, BUSTER would ask for the missing information. The state network is obviously more complex than corresponding state networks in the TABOR case. We did some informal tests with users that had achieved a high success rate and a corresponding low turn error rate on TABOR. Not surprisingly, the recogniser performance turned out to be somewhat lower than for the TABOR case. In addition to the same kind of errors as for the TABOR dialogue (like wrong bus stop), errors which led to a wrong semantic interpretation was introduced; i.e. exchange of arrival and departure names.

Thus it was concluded that the recogniser performance had to be improved before continuing the work of implementing a speech based version of BUSTER.

8 Concluding Remarks and Further Work

In this paper we have described a Norwegian dialogue system for bus travel information system, TABOR, and described a preliminary evaluation of it. We have compared the performance achieved by real users with a corresponding experiment using the recordings from a WoZ session. As expected, the results for the real users showed a significant deterioration with respect to both the relative number of illegal turns and the turn error rate, compared to the WoZ callers. The results indicate that the dialogue structure as well as the speech recognition engine should be further optimised. However, the frequent use of the system by serious but naïve users indicates that such systems are of commercial interest.

A mixed initiative dialogue system in the same domain was implemented for text based inputs. Preliminary experiments with a speech based version showed that the recogniser has to be improved in order for the system to function satisfactorily.

Currently we have just started a new project dealing with open multimodal dialogues, which should allow spontaneous speech. Hopefully this will result in a recogniser which will be able to cope with BUSTER-like dialogues.

Acknowledgements

This work was financed by the Norwegian Research Council and Telenor R&D. In addition, Telenor R&D has developed the software li-

brary, TabuLib, which forms the basis for the TABOR demonstrator. The flexible speech recogniser design procedure is due to COST Action 249 Continuous Speech Recognition over the Telephone.

References

- 1 Walker, M, Kamm, C, Boland, J. Developing and testing general models of spoken dialogue system performance. In: *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*, Athens, Greece, 189–192, May 2000.
- 2 Souvignier, B et al. The thoughtful elephant: Strategies for spoken dialog systems. *IEEE Trans. on Speech and Audioprocessing*, 8 (1), 51–62, 2000.
- 3 Amble, T. BusTUC—a natural language bus oracle. In: *Applied Natural Language Processing Conference*, Seattle, USA, Apr. 2000.
- 4 *SpeechDat*. 2003, June 27 [online] – URL: <http://www.phonetik.uni-muenchen.de/Forschung/SpeechDat/SpeechDat.html>.
- 5 Kvale, K. Amdal, I. Improved automatic recognition of natural Norwegian numbers by incorporating phonetic knowledge. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, Munich, Germany, 1763–1766, IEEE, Apr. 1997.
- 6 Odell, J et al. *The HAPI Book V1.4*. Entropic Ltd., 1999.
- 7 Johansen, F T et al. The COST 249 SpeechDat multilingual reference recogniser. In: *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*, Athens, Greece, 1351–1354, May 2000.
- 8 Höge, H et al. SpeechDat multilingual speech databases for teleservices: Across the finish line. In: *Proc. European Conf. on Speech Commun. and Techn. (EURO-SPEECH)*, Budapest, Hungary, 2699–2702, Sept. 1999.
- 9 Meliani, R E, O’Shaughnessy, D. New efficient fillers for unlimited word recognition and keyword spotting. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, Philadelphia, USA, 590–593, Oct. 1996.
- 10 Johnsen, M H et al. TABOR – A Norwegian Spoken Dialogue System for Bus Travel Information. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, Beijing, China, Oct. 2000.