Chapter 4

Speech communication and multimodal interfaces

Björn Schuller, Markus Ablaßmeier, Ronald Müller, Stefan Reifinger, Tony Poitschke, Gerhard Rigoll

Within the area of advanced man-machine interaction, speech communication has always played a major role for several decades. The idea of replacing the convential input devices such as buttons and keyboard by voice control and thus increasing the comfort and the input speed considerably, seems that much attractive, that even the quite slow progress of speech technology during those decades could not discourage people from pursuing that goal. However, nowadays this area is in a different situation than in those earlier times, and these facts shall be also considered in this book section: First of all, speech technology has reached a much higher degree of maturity, mainly through the technique of stochastic modeling which shall be briefly introduced in this chapter. Secondly, other interaction techniques became more mature, too, and in the framework of that development, speech became one of the preferred modalities of multimodal interaction, e.g. as ideal complementary mode to pointing or gesture. This shall be also reflected in the subsection on multimodal interaction. Another relatively recent development is the fact that speech is not only a carrier of linguistic information, but also one of emotional information, and emotions became another important aspect in today's advanced man machine interaction. This will be considered in a subsection on affective computing, where this topic is also consequently investigated from a multimodal point of view, taking into account the possibilities for extracting emotional cues from the speech signal as well as from visual information. We believe that such an integrated approach to all the above mentioned different aspects is appropriate in order to reflect the newest developments in that field.

4.1 Speech recognition

This section is concerned with the basic principles of *Automatic Speech Recognition (ASR)*. This research area has had a dynamic development during the last decades, and has been considered in its early stage as an almost unsolvable problem, then went through several evolution steps during the 1970s and 1980s, and eventually became more mature during the 1990s, when extensive databases and evaluation schemes became available that clearly demonstrated the superiority of stochastic machine learning techniques for this task. Although the existing technology is still far from being perfect, today there is a speech recognition market with a number of existing products that are almost all based on the before mentioned technology.

Our goal here is neither to describe the entire history of this development, nor to provide the reader with a detailed presentation of the complete state-of-the-art of the fundamental principles of speech recognition (which would probably require a separate book). Instead, the aim of this section is to present a relatively compact overview on the currently most actual and successful method for speech recognition, which is based on a probabilistic approach for modeling the production of speech using the technique of *Hidden-Markov-Models (HMMs)*. Today, almost every speech recognition system, including laboratory as well as commercial systems, is based on this technology and therefore it is useful to concentrate in this book exclusively on this approach. Although this method makes use of complicated mathematical foundations in probability theory, machine learning and information theory, it is possible to describe the basic functionality of this approach using only a moderate amount of mathematical expressions which is the approach pursued in this presentation.

4.1.1 Fundamentals of Hidden Markov Model-based Speech Recognition

The fundamental assumption of this approach is the fact that each speech sound for a certain language (more commonly called phoneme) is represented as a Hidden-Markov-Model, which is nothing else but a stochastic finite state machine.

Similarly to classical finite state machines, an HMM consists of a finite number of states, with possible transitions between those states. The speech production process is considered as a sequence of discrete acoustic events, where typically each event is characterized by the production of a vector of features that basically describe the produced speech signal at the equivalent discrete time step of the speech production process. At each of these discrete time steps, the HMM is assumed to be in one of its states, where the next time step will let the HMM perform a transition into a state that can be reached from its current state according to its topology (including possible transitions back to its current state or formerly visited states).

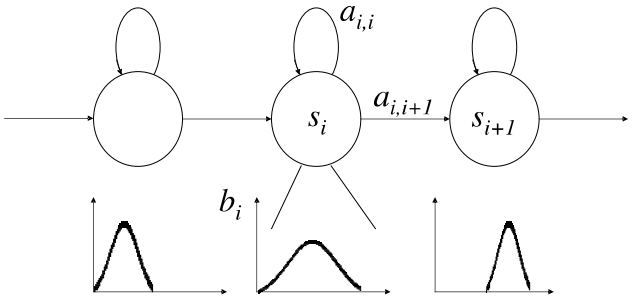

Fig. 4.1. Example for a Hidden Markov Model

Fig. 4.1 shows such an HMM, which has two major sets of parameters: The first set is the matrix of transition probabilities describing the probability $p(s(k-1) \rightarrow$

$s(k)$), where k is the discrete time index and s the notation for a state. These probabilities are represented by the parameters a in Fig. 4.1. The second set represents the so-called emission probabilities $p(x|s(k))$, which is the probability that a certain feature vector can occur while the HMM is currently in state s at time k. This probability is usually expressed by a continuous distribution function, denoted as function b in Fig. 4.1, which is in many cases a mixture of Gaussian distributions. If a transition into another state is assumed at discrete time k with the observed acoustic vector $x(k)$, it is very likely that this transition will be into a state with a high emission probability for this observed vector, i.e. into a state that represents well the characteristics of the observed feature vector.

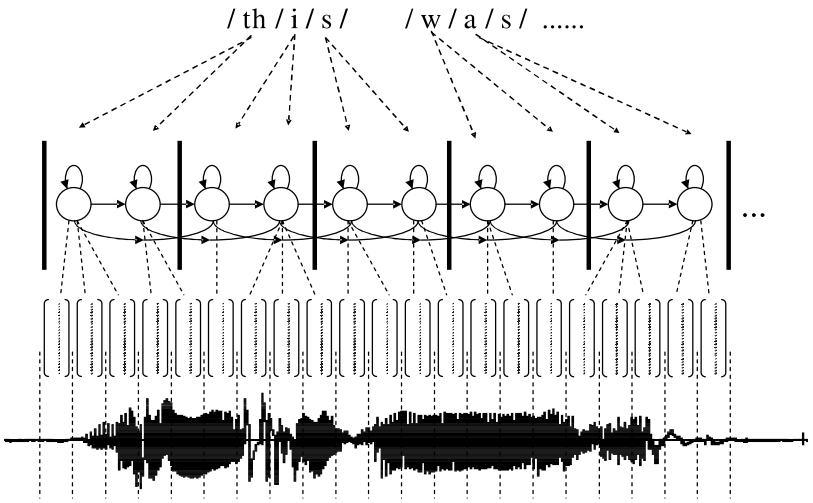

Fig. 4.2. Example for HMM-based speech recognition and training

With these basic assumptions, it is already possible to formulate the major principles of HMM-based speech recognition, which can be best done by having a closer look at Fig. 4.2: In the lower part of this figure, one can see the speech signal of an utterance representing the word sequence "this was", also displayed in the upper part of this figure. This signal has been subdivided into time windows of constant length (typically around 10 ms length) and for each window a vector of features has been generated, e.g. by applying a frequency transformation to the signal, such as a Fourier-transformation or similar. This results in a sequence of vectors displayed just above the speech signal. This vector sequence is now "observed" by the Markov-Model in Fig. 4.2, which has been generated by representing each phoneme of the underlying word sequence by a 2-state-HMM and concatenating the separate phoneme HMMs into one larger single HMM. An important issue in Fig. 4.2 is represented by the black lines that visualize the assignment of each feature vector to one specific state of the HMM. We have thus as many assignments as there are vectors in the feature vector sequence $X = [x(1), x(2), ..., x(K)]$, where K is the number of vectors,

i.e. the length of the feature vector sequence. This so-called state-alignment of the feature vectors is one of the essential capabilities of HMMs and there are different algorithms for computation of this alignment, which shall not be presented here in detail. However, the basic principle of this alignment procedure can be made clear by considering one single transition of the HMM at discrete time k from state $s(k-1)$ to state $s(k)$, where at the same time the occurrence of feature vector $x(k)$ is observed. Obviously, the joint probability of this event, consisting of the mentioned transition and the occurrence of vector $x(k)$ can be expressed according to Bayes law as:

$$\begin{aligned} p(x(k), s(k-1) \rightarrow s(k)) &= p(x(k)|s(k-1) \rightarrow s(k)) \cdot p(s(k-1) \rightarrow s(k)) \\ &= p(x(k)|s(k)) \cdot p(s(k-1) \rightarrow s(k)) \end{aligned} \tag{4.1}$$

and thus is exactly composed out of the two parameter sets a and b mentioned before, that describe the HMM and are shown in Fig. 4.1 As already stated before, a transition into one of the next states will be likely, that results into a high joint probability as expressed in the above formula. One can thus imagine that an algorithm for computing the most probable state sequence as alignment to the observed feature vector sequence must be based on an optimal selection of a state sequence that eventually leads to the maximization of the product of all probabilities according to the above equation for $k = 1$ to K. A few more details on this algorithm will be provided later.

For now, let us assume that the parameters of the HMMs are all known and that the optimal state sequence has been determined as indicated in Fig. 4.2 by the black lines that assign each vector to one most probable state. If that is the case, then this approach has produced two major results: The first one is a segmentation result that assigns each vector to one state. With this, it is for instance possible to determine which vectors - and thus which section of the speech signal - has been assigned to a specific phoneme, e.g. to the sound $/i/$ in Fig. 4.2 (namely the vectors nr. 6-9). The second result will be the before mentioned overall probability, that the feature vector sequence representing the speech signal has been assigned to the optimal state sequence. Since this probability will be the maximum possible probability and no other state sequence will lead to a larger value, this probability can be considered as the overall production probability that the speech signal has been produced by the underlying hidden Markov model shown in Fig. 4.2. Thus, an HMM is capable of processing a speech signal and producing two important results, namely a segmentation of the signal into subordinate units (such as e.g. phonemes) and the overall probability that such a signal can have been produced at all by the underlying probabilistic model.

4.1.2 Training of Speech Recognition Systems

These results can be exploited in HMM-based speech recognition in the following manner: In the training phase, the HMM parameters in Fig. 4.2 are not known, but the transcription of the corresponding word sequence (here: ”what is” will be known). If some initial parameters for the large concatenated HMM of Fig. 4.2 are assumed

for the start of an iterative procedure, then it will be of course possible to compute with these parameters the optimal corresponding state sequence, as outlined before. This will be certainly not an optimal sequence, since the initial HMM parameters might not have been selected very well. However, it is possible to derive from that state sequence a new estimation for the HMM parameters by simply exploiting its statistics. This is quite obvious for the transition probabilities, since one only needs to count the occurring transitions between the states of the calculated state sequence and divide that by the total number of transitions. The same is possible for the probabilities $p(x(k)|s(k))$ which can basically be (without details) derived by observing which kind of vectors have been assigned to the different states and calculating statistics of these vectors, e.g. their mean values and variances. It can be shown that this procedure can be repeated iteratively: With the HMM parameters updated in the way as just described, one can now compute again a new improved alignment of the vectors to the states and from there again exploit the state sequence statistics for updating the HMM parameters. Typically, this leads indeed to a useful estimation of the HMM parameters after a few iterations. Moreover, at the end of this procedure, another important and final step can be carried out: By ”cutting” the large concatenated HMM in Fig. 4.2 again into the smaller phoneme-based HMMs, one obtains a single HMM for each phoneme which has now estimated and assigned parameters that represent well the characteristics of the different sounds. Thus, the single HMM for the phoneme $/i/$ in Fig. 4.2 will have certainly different parameters than the HMM for the phoneme $/a/$ in this figure and serves well as probabilistic model for this sound.

This is basically the training procedure of HMMs and it becomes obvious, that the HMM technology allows the training of probabilistic models for each speech unit (typically the unit ”phoneme”) from the processing of entire long sentences without the necessity to phoneme-label or to pre-segment these sentences manually, which is an enormous advantage.

4.1.3 Recognition Phase for HMM-based ASR Systems

Furthermore, Fig. 4.2 can also serve as suitable visualization for demonstrating the recognition procedure in HMM-based speech recognition. In this case - contrary to the training phase - now the HMM parameters are given (from the training phase) and the speech signal represents an unknown utterance, therefore the transcription in Fig. 4.2 is now unknown and shall be reconstructed by the recognition procedure. From the previous description, we have to recall once again that very efficient algorithms exist for the computation of the state-alignment between the feature vector sequence and the HMM-states, as depicted by the black lines in the lower part of Fig. 4.2. So far we have not looked at the details of such an algorithm, but shall go into a somewhat more detailed analysis of one of these algorithms now by looking at Fig. 4.3, which displays in the horizontal direction the time axis with the feature vectors x appearing at discrete time steps and the states of a 3-state HMM on the vertical axis.

Assuming that the model starts in the first state, it is obvious that the first feature vector will be assigned to that initial state. Looking at the second feature vector,

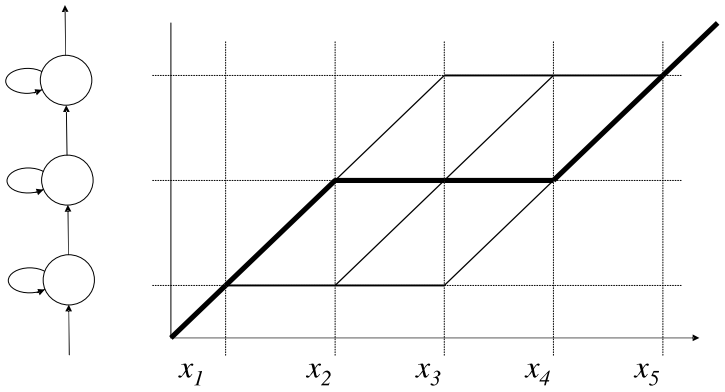

Fig. 4.3. Trellis diagram for the Viterbi-algorithm

according to the topology of the HMM, it is possible that the model stays in state 1 (i.e. makes a transition from state 1 to state1) or moves with a transition from state 1 into state 2. For both options, the probability can be computed according to Eqn. 4.1 and both options are shown as possible path in Fig. 4.3. It is then obvious that from both path end points for time step 2, the path can be extended, in order to compute if the model has made a transition from state 1 (into either state 1 or state 2) or a transition from state 2 (into either state 2 or state 3). Thus, for the 3rd time step, the model can be in state 1, 2 or 3, and all these options can be computed with a certain probability, by multiplying the probabilities obtained for time step 2 by the appropriate transition and emission probabilities for the occurrence of the third feature vector. In this way it should be easily visible that the possible state sequence can be displayed in a grid (also called trellis) as shown in Fig. 4.3, which displays all possible paths that can be taken from state 1 to the final state 3 in this figure, by assuming the observation of five different feature vectors. The bold line in this grid shows one possible path through this grid and it is clear that for each path, a probability can be computed that this path has been taken, according to the before described procedure of multiplying the appropriate production probabilities of each feature vector according to Eqn. 4.1. Thus, the optimal path can be computed with the help of the principle of dynamic programming, which is well-known from the theory of optimization. The above procedure describes the Viterbi-algorithm, which can be considered as one of the major algorithms for HMM-based speech recognition. As already mentioned several times, the major outcome of this algorithm is the optimal state sequence as well as the ”production probability” for this sequence, which is as well the probability that the given HMM has produced the associated feature vector sequence X. Let’s assume that the 3-state HMM in Fig. 4.3 represents a phoneme, as mentioned in the previous section on the training procedure, resulting in trained parameters for HMMs which typically represent all the phonemes in a given language. Typically, an unknown speech signal will either represent a spoken word or an entire spoken sentence. How can such a sentence then be recognized by the Viterbi-algorithm as described above for the state-alignment procedure of an HMM representing a single

phoneme? This can be achieved by simply extending the algorithm so that it computes the most likely sequence of phoneme-HMMs that maximize the probability of emitting the observed feature vector sequence. That means that after the final state of the HMM in Fig. 4.3 has been reached (assuming a rather long feature vector sequence of which the end is not yet reached) another HMM (and possibly more) will be appended and the algorithm is continued until all feature vectors are processed and a final state (representing the end of a word) is obtained. Since it is not known which HMMs have to be appended and what will be the optimal HMM sequence, it becomes obvious that this procedure implies a considerable search problem and this process is therefore also called ”decoding”. There are however several ways to support this decoding procedure, for instance by considering the fact that the phoneme order within words is rather fixed and variation can basically only happen between word boundaries. Thus the phoneme search procedure is more a word search procedure which will be furthermore assisted by the so-called language model, that will be discussed later and assigns probabilities for extending the search path into a new word model if the search procedure has reached the final state of a preceeding word model. Therefore, finally the algorithm can compute the most likely word sequence and the final result of the recognition procedure is then the recognized sentence. It should be noted that due to the special capabilities of the HMM approach, this decoding result can be obtained additionally with the production probability of that sentence, and with the segmentation result indicating which part of the speech signal can be assigned to the word and even to the phoneme boundaries of the recognized utterance. Some extensions of the algorithms make it even possible to compute the N most likely sentences for the given utterance (for instance for further processing of that result in a semantic module), indicating once again the power and elegance of this approach.

4.1.4 Information Theory Interpretation of Automatic Speech Recognition

With this background, it is now possible to consider the approach to HMM-based speech recognition from an information theoretic point of view, by considering Fig. 4.4.

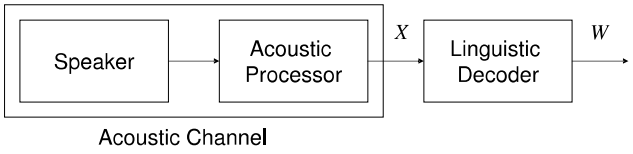

Fig. 4.4. Information theory interpretation of automatic speech recognition

Fig. 4.4 can be interpreted as follows: A speaker formulates a sentence as a sequence of words denoted as $W = [w(1), w(2), ..., w(N)]$. He speaks that sentence into the microphone that captures the speech signal which is actually seen by the automatic speech recognizer. Obviously, this recognizer does not see the speaker’s

originally uttered word sequence, but instead sees the ”encoded” version of that in form of the feature vector sequence X that has been derived from the acoustic waveform, as output of the acoustic channel as displayed in Fig. 4.4. There is a probabilistic relation between the word sequence W and the observed feature vector sequence X, and indeed this probabilistic relation is modeled by the Hidden Markov Models that represent the phoneme sequence implied by the word sequence W. In fact, this model yields the already mentioned ”production probability” that the word sequence W, represented by the appropriate sequence of phoneme HMMs has generated the observed feature vector sequence and this probability can be denoted as $p(X|W)$. The second part of Fig. 4.4 shows the so-called ”linguistic decoder”, a module that is responsible for decoding the original information W from the observed acoustic sequence X, by taking into account the knowledge about the model provided by the HMMs expressed in $p(X|W)$. The decoding strategy of this module is to find the best possible word sequence W from observing the acoustic feature string X and thus to maximize $p(W|X)$ according to Bayes' rule as follows:

$$\max_W p(W|X) = \max[p(X|W) \cdot \frac{p(W)}{p(X)}] \tag{4.2}$$

And because finding the optimal word sequence W is independent of the probability $p(X)$, the final maximization rule is:

$$\max_W [p(X|W) \cdot p(W)] \tag{4.3}$$

Exactly this product of probabilities has to be maximized during the search procedure described before in the framework of the Viterbi-algorithm. In this case, it should be noted that $p(X|W)$ is nothing else but the probability of the feature vector sequence X under the assumption that it has been generated by the underlying word sequence W, and exactly this probability is expressed by the HMMs resulting from the concatenation of the phoneme-based HMMs into a model that represents the resulting word string. In this way, the above formula expresses indeed the before mentioned decoding strategy, namely to find the combination of phoneme-based HMMs that maximize the corresponding emission probability. However, the above mentioned maximization rule contains an extra term $p(W)$ that has to be taken into account additionally to the so far known maximization procedure: This term $p(W)$ is in fact the ”sentence probability” that the word sequence W will occur at all, independent of the acoustic observation X. Therefore, this probability is described by the so-called language model, that simply expresses the occurrence probability of a word in a sentence given its predecessors, which can be expressed as:

$$p(w(n)|w(n-1), w(n-2), ..., w(n-m)) \tag{4.4}$$

In this case, the variable m denotes the ”word history”, i.e. the number of predecessor words that are considered to be relevant for the computation of the current word's appearance probability. Then, the overall sentence probability can be expressed as the product of all single word probabilities in a sentence according to

$$p(W) = \prod_{n=1}^{N} p(w(n)|w(n-1), ..., w(n-m)) \tag{4.5}$$

where N is the length of the sentence and m is the considered length of the word history. As already mentioned, the above word probabilities are completely independent of any acoustic observation and can be derived from statistics obtained e.g. from the analysis of large written text corpora by counting the occurrence of all words given a certain history of word predecessors.

4.1.5 Summary of the Automatic Speech Recognition Procedure

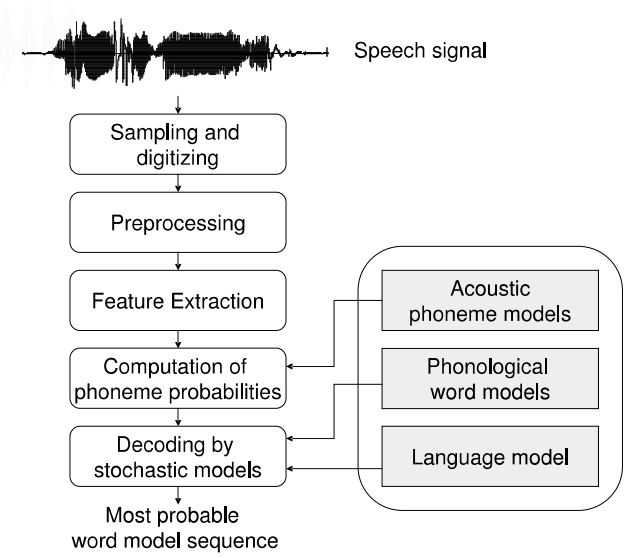

Fig. 4.5. Block diagram for HMM-based speech recognition

Finally, to summarize the functioning of HMM-based speech recognition, the block diagram in Fig. 4.5 can be interpreted as follows: The speech signal is captured by a microphone, sampled and digitized. Preprocessing of the speech signal includes some filtering process and possible noise compensation. The next step in Fig. 4.5 is feature extraction, where the signal is split into windows of roughly 10 msec length and for each window, a feature vector is computed that typically represents the windowed signal in the frequency domain. Then, in recognition mode, for each vector of the resulting feature vector sequence, state conditional probabilities are computed, basically by inserting the feature vector $x(k)$ into the right hand side of Eqn. 4.1 for each state which will be considered in the decoding procedure. According to Fig. 4.5, the gray-shaded table labeled as ”acoustic phoneme models”

contains the parameters of the distribution functions that are used to compute these state conditional probabilities. This computation is integrated into the already mentioned search procedure, that attempts to find the best possible string of concatenated HMM phoneme models that maximize the emission probability of the feature vector sequence. This search procedure is controlled by the gray-shaded table containing the phonological word models (i.e. how a word is composed of phonemes) and the table containing the language model that delivers probabilities for examining the next likely word if the search procedure has reached the final HMM-state of a previous word model. In this way, the above mentioned computation of state conditional probabilities does not need to be carried out for all possible states, but only for the states that are considered to be likely by the search procedure. The result of the search procedure is the most probable word model sequence that is displayed as transcription representing the recognized sentence to the user of the speech recognition system.

This brief description of the basic functionality of HMM-based speech recognition can of course not cover this complicated subject in sufficient detail and it is therefore not amazing that many interesting sub-topics in ASR have not been covered in this introduction. These include e.g. the area of discriminative training techniques for HMMs, different HMM architectures such as discrete, hybrid and tied-mixture approaches, the field of context-dependent modeling where acoustic models are created that model the coarticulation effects of phonemes in the context of neighboring phonemes, as well as clustering techniques that are required for representing the acoustic parameters of these extended models. Other examples include the use of HMM multi-stream techniques to handle different acoustic feature streams or the entire area of efficient decoding techniques, e.g. the inclusion of higher level semantics in decoding, fast search techniques, efficient dictionary structures or different decoder architectures such as e.g. stack decoders. The interested reader is advised to study the available literature on these topics and the large number of conference papers describing those approaches in more detail.

4.1.6 Speech Recognition Technology

As already mentioned, the HMM-technology has become the major technique for Automatic Speech Recognition and has nowadays reached a level of maturity that has led to the fact that this is not only the dominating technology in laboratory systems but in commercial systems as well. The HMM technology is also that much flexible that it can be deployed for almost every specialization in ASR. Basically, one can distinguish the following different technology lines: Small-vocabulary ASR systems with 10-50 word vocabulary, in speaker-independent mode, mainly used for telephone applications. Here, the capability of HMMs to capture the statistics of large training corpora obtained from many different speakers is exploited. The second line is represented by speaker-independent systems with medium size vocabulary, often used in automotive or multimedia application environments. Here, noise reduction technologies are often combined with the HMM framework and the efficiency of HMMs for decoding entire sequences of phonemes and words are exploited for the recognition of continuously spoken utterances in adverse environments. The last ma-

jor line are dictation systems with very large vocabulary (up to 100,000 words) which often operate in speaker-dependent and/or speaker-adaptive mode. In this case, the special capabilities of HMMs are in the area of efficient decoding techniques for searching very large trellis spaces, and especially in the field of context-dependent acoustic modeling, where coarticulation effects in continuous speech can be efficiently modeled by so-called triphones and clustering techniques for their acoustic parameters. One of the most recent trends is the developments of so-called embedded systems, where medium to large vocabulary size ASR systems are implemented on systems such as mobile phones, thin clients or other electronic devices. This has become possible with the availability of appropriate memory cards with sufficiently large storage capacity, so that the acoustic parameters and especially the memory-intensive language model with millions of word sequence probabilities can be stored directly on such devices. Due to these mentioned memory problems, another recent trend is so-called *Distributed Speech Recognition (DSR)*, where only feature extraction is computed on the local device and the features are then transferred by wireless transmission to a large server where all remaining recognition steps are carried out, i.e. computation of emission probabilities and the decoding into the recognized word sequence. For these mentioned steps, an arbitrarily large server can be employed, with sufficient computation power and memory for large language models and a large number of Gaussian parameters for acoustic modeling.

4.1.7 Applications of ASR Systems

It is not completely amazing that the above mentioned algorithms and available technologies have led to a large variety of interesting applications. Although ASR technology is far from being perfect, the currently achievable performance is in many cases satisfactory enough in order to create novel ideas for application scenarios or revisit already established application areas with now improved technology. Although current speech recognition applications are manifold, a certain structure can be established by identifying several major application areas as follows:

Telecommunications: This is still the probably most important application area due to the fact that telecommunications is very much concerned with telephony applications involving naturally speech technology and in this case, speech recognition is a natural bridge between telecommunications and information technology by providing a natural interface in order to enter data via a communication channel into information systems. Thus, most application scenarios are in the area of speech recognition involving the telephone. Prominent scenarios include inquiry systems, where the user will inquire information by calling an automated system, e.g. for banking information or querying train and flight schedules. Dialing assistance, such as speaking the telephone number instead of dialing or typing it belongs to this application area. More advanced applications include telephony interpretation, i.e. the automatic translation of phone calls between partners from different countries, and all techniques involving mobile communications, such as embedded speech recognition on mobile clients and distributed speech recognition.

Office Automation: Similarly to telecommunications, office automation has been a traditional application area of ASR for several decades and has been one of the driving forces for very large vocabulary ASR. This area has been revived by the latest emerging commercial large vocabulary speech recognition systems that really provided the required performance in terms of vocabulary size and appropriate recognition performance. Typical concrete applications in that area include the classical dictation task where a secretary creates a letter directly via voice input and other scenarios, such as e.g. the use of ASR in CAD applications or the direct command input for PCs.

Medical Applications: This area has also a quite long tradition and has been a favorite experimental scenario for ASR since at least 20 years. The most prominent field has been radiology, where the major idea has been to create a medical report directly from the visual analysis of an x-ray image by dictating directly into a microphone connected to an ASR system. Other scenarios include the control of microscopes via voice and another major application area of speech technology in general, namely the area of handicapped users, where the malfunction of hands or arms can be compensated by voice control of devices and interfaces. The size of this potential user group and the variety of different applications for handicapped users make this one of the most important ASR application scenarios.

Production and Manufacturing: This area may be less popular than the previously mentioned application areas, but has been also investigated for a long time as potentially very interesting application with a large industrial impact. Popular applications include data distribution via spoken ID codes, programming of NC machines via voice, or spoken commands for the control of large plants, such as power or chemical plants.

Multimedia Applications: Naturally, the rise of multimedia technology has also increased the demand for speech-based interfaces, as one major modality of multimodal interfaces. Most of those systems are still in the experimental phase, but some industrial applications are already underway, such as e.g. in smart environments or information kiosks, that require e.g. user input via voice and pointing. Another interesting area in this field are voice-enabled web applications, where speech recognition is used for the access to web documents.

Private Sector: This field contains several very popular speech application fields with huge potential, such as the automotive sector, electronic devices and games. Without any doubt, those represent some of the most important application fields, where consumers are ready to make some extra investment in order to add more functionality to their user interface, e.g. in case of luxury automobiles or expensive specialized electronic devices.

This overview demonstrates the huge potential of speech recognition technology for a large variety of interesting applications that can be well subdivided into the above mentioned major application areas which will represent also in the future the most relevant domains for even further improved ASR systems to come.

4.2 Speech dialogs

4.2.1 Introduction

It is strongly believed that, following command line interfaces being popular in the years 1960-80 and graphical user interfaces in the years 1980-2000, the future lies in speech and multimodal user interfaces. A lot of factors thereby clearly speak for speech as interaction form:

- Speech is the most natural communication form between humans.
- Only limited space is required: a microphone and a loudspeaker can be placed even in wearable devices. However, this does not respect computational effort.
- Hands and eyes are left free, which makes speech interaction the number one modality in many controlling situations as driving a car.
- Approximately 1.3 billion telephones exist worldwide, which resembles more than five times the number of computers connected to the Internet at the time. This provides a big market for automatic dialog systems in the future.

Using speech as an input and output form leads us to *dialog systems*: in general these are, in hierarchical order by complexity, systems that allow for *control* e.g. of functions in the car, *information retrieval* as flight data, *structured transactions* by voice, for example stock control, *combined tasks* of information retrieval and transactions as booking a hotel according to flight abilities and finally *complex tasks* as care of elder persons. More specifically a dialog may be defined as an exchange of different aspects in a reciprocal conversation between at least two instances which may either be human or machine with at least one change of the speaker.

Within this chapter we focus on spoken dialog, still it may also exist in other forms as textual or combined manners. More concretely we will deal with so called *Spoken Language Dialog Systems*, abbreviated *SLDS*, which in general are a combination of speech recognition, natural language understanding, dialog act generation, and speech synthesis. While it is not exactly defined which parts are mandatory for a SLDS, this contributes to their interdisciplinary nature uniting the fields of speech processing, dialog design, usability engineering and process analysis within the target area.

As enhancement of *Question and Answer (Q&A)* systems, which allow natural language access to data by direct answers without reference to past context (e.g. by pronominal allusions), dialog systems also allow for anaphoric expressions. This makes them significantly more natural as anaphora are frequently used as linguistic elements in human communication. Furthermore SLDS may also take over initiative. What finally distinguishes them from sheer Q&A and *Command and Control (C&C)* systems in a more complex way is that user modeling may be included. However, both Q&A and SLDS already provide an important improvement over today's predominant systems where users are asked to adhere to a given complex and artificial syntax. The following figure gives an overview of a typical circular pipeline architecture of a SLDS [20]. On top the user can be found who controls an application

found at the bottom by use of a SLDS as interaction medium. While the components are mostly the same, other architectures exist, as organization around a central process [13]. In detail the single components are:

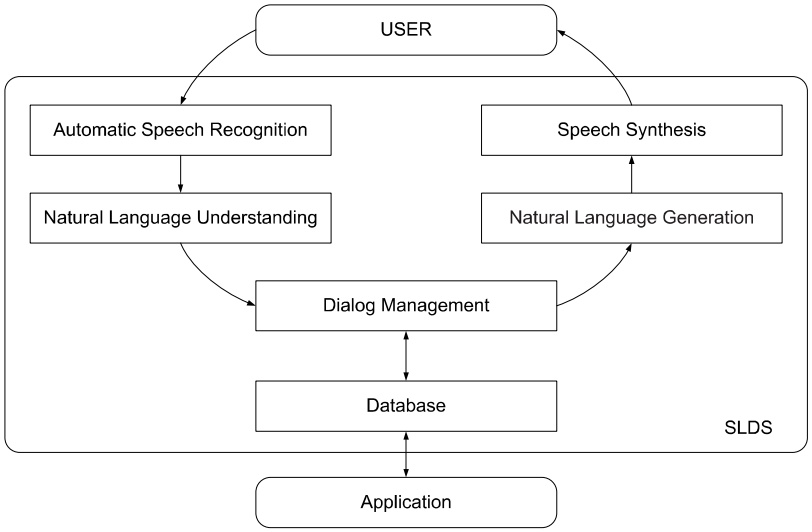

Fig. 4.6. Overview of a SLDS

- *Automatic Speech Recognition (ASR)*: Spoken input analysis leading to hypotheses of linguistic units as phonemes or words and often some form of confidence measurement of the certainty of these (see Sect. 4.1. The output is mostly provided in so called *lattices* resembling tree structures or n-best lists. Key factors of an ASR module or engine, as it is mostly referred to, are speaker(in)dependence, the vocabulary size of known words and its general robustness. On the acoustic level also a user's underlying affect may be analyzed in order to include emotional aspects (see Sect. 4.4 [45].
- *Natural Language Understanding (NLU)*: Interpretation of the intention or meaning of the spoken content. Here again several hypotheses may be forwarded to the dialog management combined with certainty.
- *Dialog Management (DM)*: The DM is arguably the central module of a voice interface as it functions as an intermediate agent between user and application and is responsible for the interaction between them. In general it operates on an intention representation provided by the NLU which models what the user (presumably) said. On the basis of this information, the DM has several options as to change the state of an underlying application in the case of voice control or retrieve a piece of data from a database of interest in the case of an information service. Furthermore the DM decides, when and which type of system voice output is performed. Shortly summarized the DM's primary tasks are storage

and analysis of context and dialog history, flow control e.g. for active initiative or barge-in handling, direction of the course of a conversation, answer production in an abstract way, and database access or application control.
- *Database (DB)*: Storage of information in respect of dialog content.
- *Natural Language Generation (NLG)*: Formulation of the abstract answer provided by the DM. A variety of approaches exists for this task reaching from probabilistic approaches with grammatical post-processing to pre-formulated utterances.
- *Speech Synthesis (TTS)*: Audio-production for the naturally formulated system answer. Such modules are in general called *Text-to-Speech* engines. The two major types of such are once *formant synthesizers* resembling a genuinely artificial production of audio by formant tract modeling and *concatenative synthesis*. Within the latter audio clips at diverse lengths reaching from phonemes to whole words of recorded speech are concatenated to produce new words or sentences. At the moment these synthesizers tend to sound more natural depending on the type of modeling and post-processing as pitch and loudness correction. More sophisticated approaches use bi- or trigrams to model phonemes in the context of their neighboring ones. Recently furthermore prosodic cues as emotional speech gain interest in the field of synthesis. However, the most natural form still remains *prerecorded speech*, while providing less flexibility or more cost at recording and storage space.

Still, as it is disputed which parts besides the DM belong to a strict definition of a SLDS, we will focus hereon in the ongoing.

4.2.2 Initiative Strategies

Before getting into dialog modeling, we want to make a classification in view of the initiative:

- *system-driven*: the system keeps the initiative throughout the whole dialog
- *user-driven*: the user keeps the initiative
- *mixed initiative*: the initiative changes throughout the dialog

Usually, the kind of application, and thus the kind of voice interface, codetermines this general dialog strategy. For example systems with limited vocabulary tend to employ rigid, *system initiative* dialogs. The system thereby asks very specific questions, and the user can do nothing else but answer them. Such a strategy is required due to the highly limited set of inputs the system can cope with. C&C applications tend to have rigid, *user initiative dialogs*: the system has to wait for input from the user before it can do anything. However, the ideal of many researchers and developers is a natural, *mixed initiative* dialog: both counterparts have the possibility of taking the initiative when this is opportune given the current state of the dialog, and the user can converse with the system as (s)he would with another human. This is difficult to obtain in general, for at least two reasons: Firstly, it is technically demanding, as the user should have the freedom to basically say anything at any moment,

which is a severe complication from a speech recognition and language processing point of view. Secondly, apart from such technical hurdles, it is also difficult from a dialog point of view, as initiative has to be tracked and reactions have to be flexible.

4.2.3 Models of Dialog

Within the DM an incorporated dialog model is responsible for the structure of the communication. We want to introduce the most important such models in the ongoing. They can be mainly divided into *structural models*, which will be introduced firstly, and *nonstructural* ones [20, 28]. Thereby a more or less predefined path is given within structural ones. While these are quite practicable in the first order, their methodology is not very principled, and the quality of dialogs based on them is arguable. Trying to overcome this, the non-structural approaches rely rather on general principles of dialog.

4.2.3.1 Finite State Model

The dialog structure represents a sequence of limited predetermined steps or states in form of a condition transition graph which models all legal dialogs within the very basic *graph-based-*, or *Finite State Automaton (FSA) model*. This graph's nodes represent system questions, system outputs or system actions, while its edges show all possible paths in the network which are labeled with the according user expressions. As all ways in a deterministic finite automaton are fixed, it cannot be spoken of an explicit dialog control. The dialog therefore tends to be rigid, and all reactions for each user-input are determined. We can denote such a model as the quintuple $\{S, s_0, s_f, A, \tau\}$. Thereby S represents a set of states besides s_0, the initial state, and s_f, the final state, and A a set of actions including an empty action ϵ. τ finally is a transition function with $\tau : S \times A \rightarrow S$. By τ it is specified to which state an action given the actual state leads. Likewise a dialog is defined as the path from s_0 to s_f within the space of possible states.

Generally, deterministic FSA are the most natural kind to represent system-driven dialogs. Furthermore they are well suitable for completely predictable information exchange due to the fact that the entire dialog model is defined at the time of the development. Often it is possible to represent the dialog flow graphically, which resembles a very natural development process.

However, there is no natural order of the conditions, as each one can be followed by any other resulting in a combinatorial explosion. Furthermore the model is unsuited for different abstraction levels of the exchanged information, and for complex dependencies between information units. Systems basing on FSA models are also inflexible, as they are completely system-led, and no path deviations are possible. The user's input is restricted to single words or phrases that provide responses to carefully designed system prompts. Tasks, which require negotiations, cannot be implemented with deterministic automata due to the uncertainty of the result. Grounding and repair is very rigid and must be applied after each turn. Later corrections by the user are hardly possible, and the context of the statements cannot be used. Summed up, such

systems are appropriate for simple tasks with flat menu structure and short option lists.

4.2.3.2 Slot Filling

In so called *frame-based systems* the user is asked questions that enable to fill slots in a template in order to perform a task [38]. These systems are more or less today's standard for database retrieval systems as flight or cinema information. In difference to automaton techniques the dialog flow is not predetermined but depends on the content of the user's input and the information that the system has to elicit. However, if the user provides more input than is requested at a time, the system can accept this information and check if any additional item is still required. A necessary component therefore is a frame that controls over already expressed information fragments. If multiple slots are yet to fill, the question of the optimal order in respect of system questioning remains. The idea thereby is to constrain the originally large set of a-priori possibilities of actions in order to speed up the dialog. Likewise, one reasonable approach to establish a hierarchy of slots is to stick to the order of highest information based on Shannon's entropy measure, as proposed in [21]. Such an item with high information might e.g. be the one leaving most alternatives open. Let therefore c_k be a random variable of items within the set C, and v_j a value of attribute a. The attribute a that minimizes the following entropy measure $H(c|a)$ will be accordingly selected:

$$H(c|a) = -\sum_{v_j \in a} \sum_{c_k \in C} P(c_k, a = v_j) \log_2 P(c_k|a = v_j) \tag{4.6}$$

Thereby the probability $P(c_k, a = v_j)$ is set to 0 unless c_k matches the revised partial description. In this case it resembles $P(c_k)$. The further missing probability $P(c_k|a = v_j)$ is calculated by the following equation, where C' is the set of items that match the revised description that includes $a = v_j$, and $P(c_k)$ can be approximated basing on the assumption that items are equally likely:

$$P(c_k|a = v_j) = \frac{P(c_k)}{\sum\limits_{c_j \in C'} P(c_j)}, P(c_k) = \frac{1}{|C|} \tag{4.7}$$

An advantage of dialog control with frames is higher flexibility for the user and multiple slot filling. The duration or dialogs are shorter, and a mixed initiative control is possible. However, an extended recognition grammar is required for more flexible user expressions, and the dialog control algorithm must specify the next system actions on basis of the available frames. The context that decides on the next action (last user input, status of the slots, simple priority ranking) is limited, and the knowledge level of the user, negotiating or collaborative planning cannot - or only be modeled to a very limited degree. No handling for communication problems exists, and the form of system questions is not specified. Still, there is an extension [50], whereby complex questions are asked if communication works well. In case of problems a system can switch to lower-level questions splitting up the high-level ones. Finally,

an overview over the system is not possible due to partially complex rules: which rule fires when?

4.2.3.3 Stochastic Model

In order to find a less hand-crafted approach which mainly bases on a designer's input, recently data-driven methods successfully applied in *machine learning* are used within the field of dialog modeling. They aim at overcoming the shortcomings of low portability to new domains and the limited predictability of potential problems within a design process. As the aim is to optimize a dialog, let us first define a cost function C. Thereby the numbers N_t of turns , N_e of errors, and N_m of missing values are therefore summed up by introduction of individual according weights w_t, w_e, and w_m:

$$C = w_t(N_t) + w_e(N_e) + w_m(N_m) \tag{4.8}$$

Now if we want to find an ideal solution minimizing this target function, we face $|A|^{|S|}$ possible strategies. A variety of suited approaches for a machine based solution to this problem exists [54]. However, we chose the predominant *Markov Decision Processes (MDP)* as a probabilistic extension to the introduced FSA following [25] herein. MDP differ from FSA, as they introduce transition probabilities instead of the functions τ. We therefore denote at a time t the state s_t, and the action a_t. Given s_t and a_t we change to state s_{t+1} with the conditional probability $P(s_{t+1}|s_t, a_t)$. Thereby we respect only events one time step behind, known as the *limited horizon Markov property*. Next, we combine this with the introduced costs, whereby c_t shall be the cost if in state s_t the action a_t is performed, with the probability $P(c_t|s_t, a_t)$. The cost of a complete dialog can likewise be denoted as:

$$C = \sum_{t=0}^{t_f} c_t, \tag{4.9}$$

where t_f resembles the instant when the final state s_f is reached. In the consequence the best action $V^*(s)$ to take within a state s is the action that minimizes over incurred and expected costs for the succeeding state with the best action within there, too:

$$V^*(s) = \arg_a \min \left[c(s,a) + \sum_{s'} P_{t_f}(s'|s,a) V^*(s') \right] \tag{4.10}$$

Given all model parameters and a finite state space, the unique function $V^*(s)$ can be computed by value interaction techniques. Finally, the optimal strategy ψ^* resembles the chain of actions that minimizes overall costs. In our case of a SLDS the parameters are not known in advance, but have to be learned by data. Normally such data might be acquired by test users operating on a simulated system known as *Wizard-of-Oz* experiments [30]. However, as loads of data are needed, and there is no data like more data, the possibility of simulating a user by another stochastic process as a solution to sparse data handling exists [24]. Basing on annotated dialogs with

real users, probabilities are derived with which a simulated user acts responding to the system. Next reinforced learning e.g. by *Monte Carlo* with exploring starts is used to obtain an optimal state-action value function $Q^*(s,a)$. It resembles the costs to be expected of a dialog starting in state s, moving to s' by a, and optimally continuing to the end:

$$Q^*(s,a) = c(s,a) + \sum_{s'} P_{t_f}(s'|s,a) \arg_a \min Q^*(s',a'), \tag{4.11}$$

Note that $V^*(s) = \arg_a \min Q^*(s,a)$. Starting in an arbitrary set of $Q^*(s,a)$ the algorithm iteratively finds the costs for a dialog session. In [24] it is shown that it converges to an optimal solution, and that it is expensive to immediately shortcut a dialog by *"bye!"*.

The goals set at the beginning of this subsection can be accomplished by stochastic dialog modeling. The necessity of data seems one drawback, while it often suffices to have very sparse data, as low accuracies of the initial transition probabilities may already lead to satisfying results.

Still, the state space is hand-crafted, which highly influences the learning process. Likewise the state space itself should be learned. Furthermore the cost function determination is not trivial, especially assigning the right weights, while also highly influencing the result. In the suggested cost function no account is taken for subjective costs as user satisfaction.

4.2.3.4 Goal Directed Processing

The more or less finite state and action based approaches with predefined paths introduced so far are well suited for C&C and information retrieval, but less suited for *task-oriented dialogs* where the user and the system have to *cooperate* to solve a problem. Consider therefore a plan-based dialog with a user that has only sparse knowledge about the problem to solve, and an expert system. The actual task thereby mostly consists of sub-tasks, and influences the structure of the dialog [14]. A system has now to be able to *reason* about the problem at hand, the application, and the user to solve the task [47]. This leads to a *theorem prover* that derives conclusions by the laws of logic from a set of true considered axioms contained in a knowledge-base. The idea is to proof whether a subgoal or goal is accomplished, as defined by a theorem, yet. User modeling is thereby also done in form of stored axioms that consist of his competence and knowledge. Whenever an axiom is missing, interaction with the user becomes necessary, which claims for an interruptible theorem prover that is able to initiate a user directed question. This is known as the *missing axiom theory* [47]. As soon as a subgoal is fulfilled, the selection of a new one can be made according to the highest probability of success given the actual dialog state. However, the prover should be flexible enough to switch to different sub-goals, if the user actively provides new information. This process is iteratively repeated until the main goal is reached.

In the framework of *Artificial Intelligence (AI)* this can be described by the *Beliefs, Desires, and Intentions (BDI)* model [20], as shown in the following figure **??**fig:Zeichnung2). This model can be applied to *conversational agents* that have

beliefs about the current state of the world, and desires, how they want it to be. Basing on these they determine their intention, respectively goal, and build a plan consisting of actions to satisfy their desires. Note that utterances are thereby treated as (speech) *actions*.

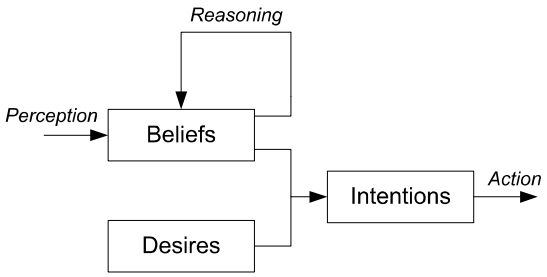

Fig. 4.7. Beliefs, Desires, and Intentions Model

To conclude, the advantages of goal directed processing are the ability to also deal with task-oriented communication and informational dialogs, a more principled approach to dialog based on a general theory of communication, and less domain-dependency at least for the general model.

Yet, by applying AI methods, several problems are inherited: high computational cost, the *frame problem* dealing with the specification of non-influenced parts of the world by actions, and the lack of proper belief and desire formalizations and independently motivated rationality principles in view of agent behavior.

4.2.3.5 Rational Conversational Agents

Rational Conversational Agents directly aim at imitation of *natural* conversation behavior by trying to overcome plan-based approaches' lack in human-like *rationality*. The latter is therefore reformulated in a formal framework establishing an intelligent system that relies on a more general competence basis [40]. The beliefs, desires and intentions are logically formulated, and a *rational* unit is constructed basing thereon that decides upon the actions to be taken.

Let us denote an agent i believing in proposition p as $B(i,p)$, and exemplary propositions as φ, and ψ. We can next construct a set of logical rules known as *NKD45* or *Weak S5* that apply for the B operator:

$$(N): \text{ always } \varphi \rightarrow \text{ always } B(i,\varphi) \tag{4.12}$$

$$(K): B(i,\varphi) \wedge B(i,\varphi \rightarrow \psi) \rightarrow B(i,\psi) \tag{4.13}$$

$$(D): B(i,\varphi) \rightarrow \neg B(i,\neg\varphi) \tag{4.14}$$

$$(4): B(i,\varphi) \rightarrow B(i,B(i,\varphi)) \tag{4.15}$$

$$(5): \neg B(i,\varphi) \rightarrow B(i,\neg B(i,\varphi) \tag{4.16}$$

Let us furthermore denote desires, respectively goals, as $G(i, p)$. Likewise an agent indexed i has the desire that p comes true. As an exercise, one can reflect why only (K) and (4) apply for goals:

$$(K) : G(i, \varphi) \wedge G(i, \varphi \rightarrow \psi) \rightarrow G(i, \psi) \tag{4.17}$$

$$(4) : G(i, \varphi) \rightarrow G(i, B(i, \varphi)) \tag{4.18}$$

We next connect goals and beliefs by a *realism constraint* which states that an agent cannot have a goal he believes to be false:

$$B(i, \varphi) \rightarrow G(i, \varphi) \tag{4.19}$$

Expected consequences of a goal furthermore also have to be a goal of an agent, which is known as the *expected consequences constraint*:

$$G(i, \varphi) \wedge B(i, \varphi \rightarrow \psi) \rightarrow G(i, \varphi) \tag{4.20}$$

Finally, let us introduce the *persistent goal constraint* with the persistent goal $PG(i, \varphi)$: an agent i should not give up a goal φ to be true in the future (1) that he believes not true presently (2) until either its fulfillment or necessarily non-availability (3).

$$(1) : PG(i, \varphi) \text{ if and only if } G(i, (\textit{Future } \varphi)) \wedge \tag{4.21}$$

$$(2) : B(i, \neg\varphi) \wedge \tag{4.22}$$

$$(3) : [\text{Before } ((B(i, \varphi) \vee B(i, (\text{Necessary } \neg\varphi))) \neg G(i, (\text{Future } \varphi)))] \tag{4.23}$$

The rational unit of a SLDS fed with the formalized notions of belief, desires and intentions has now a selection of communication actions to choose from that are associated with *feasibility preconditions* and *rational effects* [40]. Such an action is e.g. that an agent i informs an agent j about φ, whereby the user is also understood as an agent. Now, if an agent has the intention to achieve a goal, it selects an appropriate action and thereby inherits the intention to fulfill according preconditions. This approach defines a planning algorithm demanding for a theorem prover as in the previous section. However, user directed questions do not directly concern missing axioms - it is rather only checked whether preconditions are fulfilled and effects of system-actions help in view of the current goal.

To sum up, the characteristics of rational conversational agents are full mixed initiative, the ability to implement theoretically complex systems which solve dynamic and (only) cooperative complex tasks. They are also much more oriented on the linguistic theory.

However, there is no general definition of the agent term besides that they should be reactive, autonomous, social, rational and antropomorph. Furthermore way more resources are needed both quantitatively (computer speed, programming expenditure) and qualitatively (complexity of the problems which can be solved). Also, the formalization of all task relevant domain-knowledge is not trivial. So far there are only academic systems realized.

4.2.4 Dialog Design

Let us now turn our attention to the *design* of a dialog, as a lot of factors besides recognition performance of the ASR unit have significant influence on the utmost design goals *naturalness*, *efficiency*, and *effectiveness*. As we learned so far, initiative can e.g. be chosen mixed or by one side only. Furthermore confirmations can be given or spared, and suggestions made in case of failure, etc. In the following three main steps in the design of a dialog will be outlined:

- *Script writing*: In this part, also known as *call flow layout*, the interaction between user and system is laid out step-wisely. Focus should be given to the naturalness of the dialog, which is also significantly influenced by the quality of the dialog flow.
- *Prompt Design*: Similar to a prompt in console based interfaces a sound or announcement of the system signals the user *when* and depending on the situation *what* to speak. This acoustic prompt has to be well designed in order to be informative, well heard, but not disturbing. A frequent prompt might therefore be chosen short. On the other hand in the case of error handling, tutorial information or help provision, prompts might be chosen more complex, as the quality of a user's answer depends strongly on the quality of the prompt. Finally, only by appropriate prompt crafting the initiative throughout a dialog may be controlled effectively.
- *Grammar Writing*: Within the grammar the possible user statements given a dialog-state are defined. A compromise between coverage and recognition accuracy has to be found, as too broad a coverage often leads to decreased performance due to a large sphere of possible hypotheses. Also, phrases of multiple words should be handled. This is often realized by finite state grammars, and triggering by keywords.

For an automatic system it seems crucial to understand the actual meaning of a user statement, which highly depends on the context. Furthermore it is important to design system announcements clear and understandable for the user. We therefore also want to take a brief linguistic view on dialog within this section.

Three different aspects are important considering the meaning of a verbally uttered phrase in the chain from sender to receiver: Firstly, we have the *locution*, which represents the semantic or literal significance of the utterance; secondly there is the *illocution*, the actual intention of the speaker, and finally the *perlocution* stands for how an utterance is received by the counterpart. Likewise to speak is to perform a locution, but to speak with an intent (ask, promise, request, assert, demand, apologize, warn, etc.) is to perform an illocution. The purpose, the *illocutionary intent*, is meaningful and will ordinarily be recognized by hearers. Within this context we want to have a look at four well known Greek conversational maxims:

- *Maxim of relevance: be relevant*. Consider hereon: 'He kicked the bucket' (we assume that someone died because that's what's relevant), or 'Do you know what time it is?' (we assume that the speaker wants to know the time because the "real" question is irrelevant).

- *Maxim of quality: be truthful*. E.g. 'If I hear that song again I'll kill myself' (we accept this as a hyperbole and do not immediately turn the radio off), or 'The boss has lost his marbles' (we imagine a mental problem and not actual marbles).
- *Maxim of quantity: be informative, say neither too much not too little.* (Asked the date, we do not include the year).
- *Maxim of manner: be clear and orderly.*

In order to obtain high overall dialog quality, some aspects shall be further outlined: Firstly, consistency and transparency are important to enable the user to picture a model of the system. Secondly, social competence in view of user behavior modeling and providing a personality to the system seems very important. Thirdly, error handling plays a key role, as speech recognition is prone to errors. The precondition thereby is *clarification*, demanding that problems at the user-input (no input, cut input, word-recognition errors, wrong interpretations, etc.) must become aware to the system. In general it is said that users accept a maximum of five percent errors or less. Therefore trade-offs have to be made considering the vocabulary size, and naturalness of the input. However, there is a chance to raise the accuracy by expert prompt modeling, allusion to the problem nature, or at least cover errors and reduce annoyance of the users by a careful design. Also, the danger of over-modeling a dialog in view of world knowledge, social experience or general complexity of manlike communication shall be mentioned. Finally, the main characteristic of a good voice interface is probably that is usable. *Usability* thereby is a widely discussed concept in the field of interfaces, and various operationalizations have been proposed. In [30] it is stated that usability is a multidimensional concept comprising *learnability*, *efficiency*, *memorability*, *errors* and *satisfaction*, and ways are described, in which these can be measured. Defining when a voice interface is usable is one thing, developing one is quite another. It is by now received wisdom that usability design is an iterative process which should be integrated in the general development process.

4.2.5 Scripting and Tagging

We want to conclude this section with a short introduction of the two most important dialog mark-up languages in view of scripting and tagging: *VoiceXML* and *DAMSL*.

VoiceXML (VXML) is the W3C's standard XML format for specifying interactive voice dialogs between a human and a computer. VXML is fully analogous to HTML, and just as HTML documents are interpreted by a visual web browser, VXML documents are interpreted by a voice browser. A common architecture is to deploy banks of voice browsers attached to the public switched telephone network so that users can simply pick up a phone to interact with voice applications. VXML has tags that instruct the voice browser to provide speech synthesis, automatic speech recognition, dialog management, and soundfile playback. Considering dialog management, a form in VXML consists of fields and control units: fields collect information from the user via speech input or DMTF (dual tone multi-frequency). Control units are sequences of procedural statements, and the control of the dialog is made by the following form interpretation algorithm, which consists of at least one major loop with three phases:

- *Select*: The first form not yet filled is selected in a top down manner in the active VXML document that has an open guard condition.
- *Collect*: The selected form is visited, and the following prompt algorithm is applied to the prompts of the form. Next the input grammar of the form is activated, and the algorithm waits for user input.
- *Process*: Input evaluation of a form's fields in accordance to the active grammar. Filled elements are called for example to the input validation. The process phase ends, if no more items can be selected or a jump point is reached.

Typically, HTTP is used as the transport protocol for fetching VXML pages. While simpler applications may use static VXML pages, nearly all rely on dynamic VXML page generation using an application server. In a well-architected web application, the voice interface and the visual interface share the same back-end business logic.

Tagging of dialogs for machine learning algorithms on the other hand is mostly done using *Dialogue Act Mark-up in Several Layers (DAMSL)* - an annotation scheme for communicative acts in dialog [6]. While different dialogs being analyzed with different aims in mind will lead to diverse acts, it seems reasonable to agree on a common basis for annotation in order to enable database enlargement by integration of other ones. The scheme has three layers: *Forward Communicative Functions*, *Backward Communicative Functions*, and *Information Level*. Each layer allows multiple communicative functions of an utterance to be labeled. The Forward Communicative Functions consist of a taxonomy in a similar style as the actions of traditional speech act theory. The most important thereby are *statement* (assert, reassert, other statement), *influencing addressee future action* (open-option, directive (info-request, action-directive), and *committing speaker future action* (offer, commit). The Backward Communicative Functions indicate how the current utterance relates to the previous dialog: *agreement* (accept, accept-part, maybe, reject-part, reject, hold), *understanding* (signal non-understanding, signal understanding (acknowledge, repeat phrase, completion), correct misspeaking), and *answer*. Finally, the Information Level annotation encodes whether an utterance is occupied with the dialog task, the communication process or meta-level discussion about the task.

4.3 Multimodal interaction

The research field of *Human-Computer Interaction* (HCI) focuses on arranging the interaction with computers easier, safer, more effective and to a high degree seamless for the user.

> "Human-Computer Interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." [16]

As discribed, HCI is an interdisciplinary field of research where many different subjects are involved to reach the long-term objective of a natural and intuitive

way of interaction with computers. In general, the term interaction describes the mutual influence of several participants to exchange information that is transported by most diverse means in a bilateral fashion. Therefore, a major goal of HCI is to converge the interface increasingly towards a familiar and ordinary interpersonal way of interaction. For natural human communication several in- and output channels are combined in a multimodal manner.

4.3.1 In- and Output Channels

The human being is able to gather, process and express information through a number of channels. The input channel can be discribed as *sensor function* or perception, the processing of information as *cognition*, and the output channel as *motor function* (see figure 4.8).

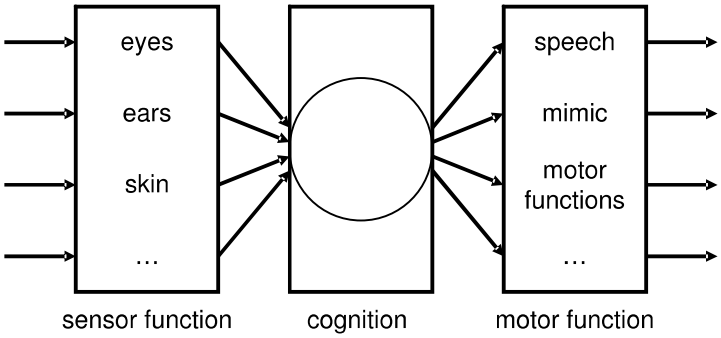

Fig. 4.8. Human information input and output channels

Humans are equipped with six senses respectively sense organs to gathers stimuli. These senses are defined by physiology [12]:

sense of sight	visual channel
sense of hearing	auditive channel
sense of smell	olfactory channel
sense of taste	gustatory channel
sense of balance	vestibular channel
sense of touch	tactile channel

The human being is also provided with broad range of abilities for information output. The *output channel* can process less amount of information than the input channel. Outgoing information is transmitted by the *auditive*, the *visual* and the *haptic* channel (tactile as perception modality is distinguished from haptic as output manner). By this means, the human is enabled to communicate with others in a simple, effective, and error robust fashion. With an integrated and synchronized use of

different channels he can flexibly adapt to the specific abilities of the conversational partner and the current surroundings.

However, HCI is far from information transmission of this intutive manner. This is due to the limitations of the technical systems in regard of their number and performance of the single in- and output modalities. The term modality refers to the type of communication channel used to convey or acquire information. One of the core difficulties of HCI is the divergent boundary conditions between computers and human. Nowadays, the user can transfer his commands to the computer only by standard input devices - e.g. mouse or keyboard. The computer feedback is carried out by visual or acoustic channel - e.g. monitor or speakers. Thus, todays' technology uses only a few interaction modalities. The term multimodality is used, whenever two or more of these modalities are involved.

4.3.2 Basics of Multimodal Interaction

The term "*multimodal*" is derived from "*multi*" (lat.: several, numerous) and "*mode*" (lat.: naming the method). The word "modal" may cover the notion of "modality" as well as that of "mode". Scientific research focuses on two central characteristics of multimodal systems:

- the user is able to communicate with the machine by several input and output modes
- the different information channels can interact in a sensible fashion

According to S. Oviatt [32] multimodal interfaces combine natural input modes - such as speech, pen, touch, manual gestures, gaze and head and body movements - in a coordinated manner with multimedia system output. They are a new class of interfaces which aims to recognize naturally occuring forms of human language and behavior, and which incorporate one or more recognition-based technologies (e.g., speech, pen, vision). Benoit [4] espanded the definition to a system, which represents and manipulates information from different human communication channels at multiple levels of abstraction. These systems are able to automatically extract meaning from multimodal raw input data and conversely produce perceivable information from symbolic abstract representations. In 1980, Bolt's "Put That There" [**?**] demonstration showed the new direction for computing which processed speech in parallel with touch-pad pointing. Multimodal systems benefit from the progress in recognition-based technologies which are capable to gather naturally occuring forms of human language and behavior. The dominant theme in users' natural organization of multimodal input actually is complementary of content that means each input consistently contributes to different semantic information. The partial information sequences must be fused and can only be interpreted altogether. However, redundancy of information is much more less common in human communication. Sometimes different modalities can input concurrent content that has to be processed independently. Multimodal applications range from map-based (e.g. tourist information) and virtual reality systems, to person identification and verification systems, to medical and web-based transaction systems. Recent systems integrate two or more

recognition-based technologies like speech and lips. A major aspect is the integration and synchronization of these multiple information streams what is discussed intensely in Sect. 4.3.3.

4.3.2.1 Advantages

Multimodal interfaces are largely inspired by the goal of supporting more transparent, flexible, effective, efficient and robust interaction [26, 32]. The flexible use of input modes is an important design issue. This includes the choice of the appropriate modality for different types of information, the use of combined input modes, or the alternate use between modes. A more detailed differentiation is made in Sect. 4.3.2.2. Input modalities can be selected according to context and task by the user or system. Especially for complex tasks and environments, multimodal systems permit the user to interact more effectively. Because there are large individual differences in abilities and preferences, it is essential to support selection and control for diverse user groups [33]. For this reason, multimodal interfaces are expected to be easier to learn and use. The continuously changing demands of mobile applications enables the user to shift these modalities, e.g. in-vehicle applications. Many studies proof that multimodal interfaces satisfy higher levels of user preferences. The main advantage is probably the efficiency gain that derives from the human ability to process input modes in parallel. The human brain structure is developed to gather a certain kind of information with specific sensor inputs. Multimodal interface design allows a superior error handling to avoid and to recover from errors which can have user-centered and system-centered reasons. Consequently, it can function in a more robust and stable manner. In-depth information about error avoidance and graceful resolution from errors is given in Sect. 4.3.4. A future aim is to interpret continuous input from visual, auditory, and tactile input modes for everyday systems to support intelligent adaption to user, task and usage environment.

4.3.2.2 Taxonomy

A base taxonomy for the classification of multimodal systems was created in the MIAMI-project [15] by Nigay and Coutaz. As a basic principle there are three decisive degrees of freedom for multimodal interaction:

- the degree of abstraction
- the manner (temporal) of application
- the fusion of the different modalities

Figure 4.9 shows the resulting classification space and the consequential four basic categories of multimodal applications dependent on the parameters value of data fusion (*combined*/*independent*) and temporal usage (*sequential*/*parallel*). The exclusive case is the simplest variant of a multimodal system. Such a system supports two or more interaction channels but there is no temporal or content related connection. A sequential application of the modalities with functional cohesion is denominated as alternative multimodality. Beside a sequential appliance of the modalities there is the possibility of parallel operation of different modalities as seen in figure 4.9. Thereby,

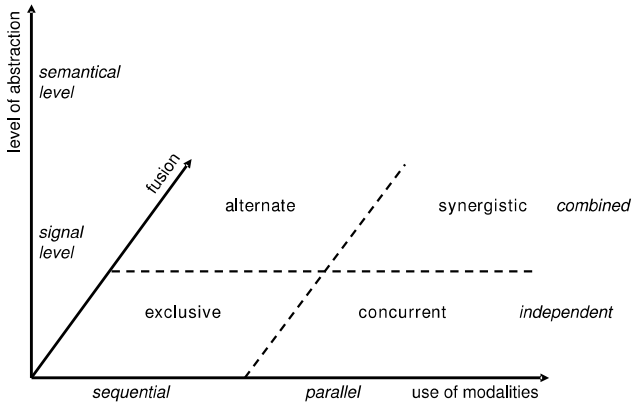

Fig. 4.9. Classification space for classification of multimodal systems [41]

it is differentiated between the manner of fusion of the interaction channels in simultaneous and synergistic multimodality. The third degree of freedom is the level of abstraction. This refers to the technical level, on which the signals are processed that ranges from simple binary sequences to highly complex semantic terms.

4.3.3 Multimodal Fusion

As described, in multimodal systems the information flow from human to computer occurs by different modalities. To be able to utilize the information transmitted by different senses, it is necessary to integrate the input channels to create an appropriate command which is equivalent to the user's intention. The input data gathered from the single modalities is generated via single mode recognizers. There are three basic approaches for combining the results of each recognition modules to one information stream [10, 53]:

- *Early (signal) fusion*: The earliest possible fusion of the sensor data is the combination of the sensor specific raw data. The classification of the data is mostly achieved by Hidden-Markov-Models (HMM), temporal Neural Networks (NN) or Dynamic Bayesian Networs (DBN). Early fusion is well suited for temporally synchronized inputs. This approach to fusion only succeeds if the data provided by different sources is of the same type and a strong correlation of modalities exists. For example the fusion of the images generated with a regular camera and an infrared camera for use in night vision systems. Furthermore, this type of fusion is applied in speech recognition systems supported by lip-reading technology, in which the viseme[1] and phoneme progression can be registered collective in one HMM. A great problem of early fusion is the large data amount

[1] A visem is the generic image of the face (especially the lip positioning) in the moment of creation of a certain sound. Visemes are thus the graphic pendant to phonemes.

necessary for the training of the utilized HMMs.

- *Late (semantic) fusion*: Multimodal systems which use late fusion consist of several single mode recognition devices as well as a downstream data fusion device. This approach contains a separate preprocessing, feature extraction and decision level for each separate modality. The results of the separate decision levels are fused to a total result. For each classification process each discrete decision level delivers a probability result respective a confidence result for the choice of a class n. These confidence results are afterwards fused for example by appropriate linear combination. The advantage of this approach is the different recognition devices being independently realizable. Therefore the acquisition of multimodal data sets is not necessary. The separate recognition devices are trained with monomodal data sets. Because of this easy integration of new recognizers, systems that use late fusion scale up easier compared to early fusion, either in number of modalities or in size of command set. [53]

- *Soft decision fusion*: A compromise between early and late fusion is the so called soft decision fusion. In this method the confidence of each classifier is also respected as well as the integration of an N-best list of each classifier.

In general, multimodal systems consist of various modules (e.g., different single mode recognizers, multimodal integration, user interface). Typically, these software components are developed and implemented independently from each other. Therefore, different requirements to the software architecture of a multimodal system arise. A common infrastructure approach that has been adopted by the multimodal research community involves multi-agent architectures, where agents are defined as any software process. In such architectures, the many modules that are needed to support the multimodal system, may be written in different programming languages and run on several machines. One example for an existing multimodal framework is given by [27]. The system architecture consists of three main processing levels: the input level, the integration level, and the output level. The input level contains any kind of interface that is capable of recognizing user inputs (e.g., mouse, buttons, speech recognizer, etc.). Dedicated command mappers (CMs) encode the information bits of the single independent modality recognizers and context sensors into a meta language based on a context-free grammar (CFG). In the integration level, the recognizer outputs and additional information of context sensors (e.g., information about application environment, user state) are combined in a late semantic fusion process. The output level provides any devices for adequate multimodal system feedback.

4.3.3.1 Integration Methods

For signal and semantic fusion, there exist different integration methods. In the following, some of these methodes are explained:

- ***Unification-based integration:***

Typed-feature structure unification is an operation that verifies the consistency of two or more representational structures and combines them into a single result. Typed feature structures are used in natural language processing and computational linguistics to enhance syntactic categories. They are very similar to frames from knowledge representation systems or records from various programming languages like C, and have been used for grammar rules, lexical entries and meaning representation. A feature structure consists of a type, which indicates the kind of entity it represents, and an associated collection of feature-value or attribute-value pairs. [Carpenter: The Logic of Typed Feature Structures] Unification-based integration allows different modalities to mutually compensate for each others' errors [M. Johnston: Unification-based Multimodal Integration]. Feature structure unification can combine complementary and redundant input, but excludes contradictory input. For this reason it is well suited to integrate e.g. multimodal speech and gesture inputs.

- ***Statistical integration:***
  Every input device produces an N-best list with recognition results and probabilities. The statistical integrator produces a probability for every meaningful combination from different N-best lists, by calculating the cross product of the individual probabilities. The multimodal command with the best probability is then chosen as the recognized command.
  In a multimodal system with K input devices we assume that $R_{i,j} = 1, \ldots, N_i$ with the probabilities $P_{i,j} = P_i(1), \ldots, P_i(N_i)$ are the N_i possible recognition results from interface number $i, i \in (1, \ldots, K)$. The statistical integrator calculates the product of each combination of probabilities

$$C_n, n = 1, \ldots, \prod_{i=1}^{K} N_i \tag{4.24}$$

  combinations with the probabilities

$$P_n = \prod_{i=1}^{K} P_{i,j}, j \in (1, \ldots, N_i) \tag{4.25}$$

  The integrator chooses which combinations represent valid system commands. The choice could be based on a semantically approach or on a database with all meaningful combinations. Invalid results are deleted from the list giving a new list $C_{valid,n}$ with at most $\prod_{i=1}^{K} N_i$ valid combinations. The valid combination with the maximum probability $max[P_{valid,n}]$ is chosen as the recognized command.
  The results can be improved significantly if empirical data is integrated in the statistical process. Statistical integrators are easy to scale up, because all application knowledge and empirical data is integrated at the configuration level. Statistical integrators will work very well with supplementary information, and good with complementary. A problem is, how the system should react, when

no suggestive combinations are available. Furthermore the system has to be programmed with all meaningful combinations to decide which ones represent valid system commands. Another possibility to decide whether a combination is valid or not is to use a semantical integration process after the statistical integration. This avoids the explicit programming of all valid combinations.

- ***Hybrid processing:***
  In hybrid architectures symbolic unification-based techniques that integrate feature structures are combined with statistical approaches. In contrast to symbolic approaches these architectures are very robust functioning. The Associative Mapping and Members-Teams-Committee (MTC) are two main techniques. They develop and optimize the following factors with a statistical approach: the mapping structure between multimodal commands and their respective constituents, and the manner of combining posterior probabilities. The Associative Mapping defines all semantically meaningful mapping relations between the different input modes. It supports a process of table lookup that excludes consideration of those feature structures that can impossibly be unified semantically. This table can be build by the user or automatically. The associative mapping approach basically helps to exclude concurrency inputs form the different recognizer and to quickly rule out impossible combinations. Members-Team-Committee is a hierarchical technique with multiple members, teams and committees. Every recognizer is represented by a member. Every member reports his results to the team leader. The team leader has an own function to weight the results and reports it to the committee. At last, the committee chooses the best team, and reports the recognition result to the system. The weightings at each level have to be trained.

- ***Rule-based integration:***
  Rule-based approaches are well established and applied in a lot of integration applications. They are similar to temporal approaches, but are not strictly bound to timing constraints. The combination of different inputs is given by rules or look-up tables. For example in a look-up table every combination is rated with a score. Redundant inputs are represented with high scores, concurrency information with negative scores. This allows to profit from redundant and complementary information and excludes concurrency information.
  A problem with rule-based integration is the complexity: many of the rules have preconditions in other rules. With increasing amount of commands these preconditions lead to an exponential complexity. They are highly specified and bound to the domains developed for. Moreover application knowledge has to be integrated in the design process to a high degree. Thus, they are hard to scale up.

- ***Temporal aspects:***
  Overlapped inputs, or inputs that fall within a specific period of time are combined by the temporal integrator. It checks if two or more signals from different input devices occur in a specific period of time. The main use for temporal in-

tegrators is to determine whether a signal should be interpreted on its own or in combination with other signals. Programmed with results from user studies the system can use the information how users react in general. Temporal integration consists of two parts: microtemporal and macrotemporal integration:
Microtemporal integration is applied to combine related information from various recognizers in parallel or in a pseudo-parallel manner. Overlapped redundant or complementary inputs are fused and a system command or a sequence of commands is generated.
Macrotemporal integration is used to combine related information from the various recognizers in a sequential manner. Redundant or complementary inputs, which do not directly overlap each other, but fall together in one timing window are fused by macrotemporal integration. The macrotemporal integrator has to be programmed with results from user studies to determine which inputs in a specified timing window belong together.

4.3.4 Errors in multimodal systems

Generally, we define an error in HCI, if the user does not reach her or his desired goal, and no coincidence can be made responsible for it. Independent from the domain, error robustness substantially influences the user acceptance of a technical system. Based on this fact, error robustness is necessary. Basically, errors can never be avoided completly. Thus, in this field both passive (a-priori error avoidance) and active (a-posteriori error avoidance) error handling are of great importance. Undesired system reactions due to faulty operation as well as system-internal errors must be avoided as far as possible. Error resolution must be efficient, transparent and robust. The following scenario shows a familiar error-prone situation: A driver wants to change the current radio station using speech command ”listen to hot radio”. The speech recognition misinterprets his command as ”stop radio” and the radio stops playing.

4.3.4.1 Error classification

For a systematic classification of error types, we basically assume that either the user or the system can cause an error in the human-machine communication process.

4.3.4.2 User specific errors

The user interacting with the system is one error source. According to J. Reason [39] user specific errors can be categorized in three levels:

- ***Errors on the skill-based level*** *(e.g., slipping from a button)*
  The skill-based level comprises smooth, automated, and highly integrated routine actions that take place without conscious attention or control. Human performance is governed by stored patterns of pre-programmed instructions represented as analog structures in a time-space domain. Errors at this level are related to the intrinsic variability of force, space, or time coordination. Sporadically, the user checks, if the action initiated by her or him runs as

planned, and if the plan for reaching the focused goal is still adequate. Error patterns on skill-based level are execution or memory errors that result from inattention or overattention of the user.

- ***Errors on the rule-based level*** *(e.g., using an valid speech command, which is not permitted in this, but in another mode)*
  Concerning errors on the rule-based level, the user violates stored prioritized rules (so-called productions). Errors are typically associated with the misclassification of situations leading to the application of the wrong rule or with the incorrect recall of procedures.

- ***Errors on the knowledge-based level*** *(e.g., using a speech command, which is unknown to the system)*
  At the knowledge-based level, the user applies stored knowledge and analytical processes in novel situations in that actions must be planned on-line. Errors at this level arise from resource limitations (bounded rationality) and incomplete or incorrect knowledge.

4.3.4.3 System specific errors

In the error taxonomy errors caused by the system are addressed. System specific errors can be distinguished in three categories:

- ***Errors on the recognition level***
  Examples are errors like misinterpretation, false recognition of a correct user input, or an incorrect system-intrinsic activation of a speech recognizer (e.g., the user coincidentally applies the keyword which activates the speech recognizer in a conversation).

- ***Errors on the processing level***
  Timing problems or contradictory recognition results of different monomodal recognizers, etc. (e.g., the result of speech recognition differs from gesture recognition input) are causing processing errors.

- ***Errors on the technical level***
  System overflow or breakdown of system components are leading to system errors.

4.3.4.4 Error avoidance

According to [46] there are eight rules for designing user interfaces. These rules are derived from experience and applicable in most interactive systems. They do not conduce error avoidance in a direct way, but they simplifiy the user's interaction with the system. In that way, many potential errors are prevented. From these rules one can derive some guidelines for multimodal interfaces:

- *Strive for consistency*: Similar situations should consist of similar sequences of action, as identical terminology e.g. menus or promts. Consistency means for multimodal interfaces consistency in two ways. First, within one situation all commands should be the same for all modalities. Second, within one modality, all similar commands in different situations should be the same. E.g., the command to return to the main menu should be the same command in all submenus and all submenus should be accessable by all modalities by the same command (menuname on the button is identical with the speech command).
- *Offer informative feedback*: Every step of interaction should be answered with a system feedback. This feedback should be modest for frequent and minor actions and major for infrequent or major actions. Multimodal interfaces have the opportunity to use the advantage of different output modalities. Feedback should be given in the same modality as the used input modality. E.g., a speech command should be answered with a acoustical feedback.
- *Reduce short-term memory load*: The human information processing is limited in short-term memory. This requires simple dialog system. Multimodal systems should use the modalities in a way, that reduces the user's memory load. E.g., object selection is easy by pointing on it, but difficult by using speech commands.
- *Synchronize multiple modalities*: Interaction by speech is highly temporal. Visual interaction is spatial. Synchronisation of these input modalities is needed. E.g., selection of objects by pointing with the finger on it and starting the selection by using speech (”Select this item”).

There are also many other design aspects and guidelines to avoid errors. In comparison to monomodal interfaces, multimodal interfaces can even improve error avoidance by enabling the user to choose freely which input modality to use. In this way, the user is able to select the input modality, which is the most comfortable and efficient way to achieve his aim. Also, if interacting by more than one channel, typical errors of a single modality can be compensated by merging all input data to one information. So, providing more than one input modality increases the robustness of the system and helps to avoid errors in advance.

4.3.4.5 Error resolution

In case of occurring errors (system or user errors) the system tries to solve upcoming problems by initiating dialogs with the user. Error resolution strategies are differentiated in single-step and multi-level dialog strategies. In the context of a single-step strategy a system promt is generated, to which the user can react with an individual input. On the other hand, in the context of a multi-level strategy, a complex further inquiry dialog is initiated, in which the user is led step by step through the error resolution process. Especially, the second approach offers enormous potential for an adaptation to the current user and the momentary environment situation.

For example, the following error handling strategies can be differentiated:

- *Warning*
- *Asking for repetition of last input*
- *Asking to change the input modality*

- *Offering alternative input modalities*

These strategies differ by characteristics as initialization of the error warning, strength of context, individual characteristics of the user, complexity of the error strategy, and inclusion of the user. The choice which dialog strategy to use depends mainly on contextual parameters and current state of the system (e.g., eventually chosen input modality, state of the application). According to Sect. **??**, the error management component is located in the integration level of a multimodal architecture. The error management process consists of four steps: error feature extraction, error analysis, error classification, and error resolution. First, a module continuously extracts certain features from the stream of incoming messages and verifies an error potential. Then, the individual error patterns can be classified. In this phase of the error management process, the resulting error type(s) are determined. Afterwards, for the selection of a dedicated dialog strategy, the current context parameters as well as the error types are analyzed. From the results, the strategy with the highest plausibility is chosen and finally helps to solve the failure in an comfortable way for the user. Sumarizing, the importance of error robustness for multimodal systems has been discussed and ways of avoidance and resolution have been presented.

4.4 Emotions from speech and facial expressions

Today the great importance of the integration of emotional aspects as the next step toward more natural human-machine interaction is commonly accepted. Throughout this chapter we therefore want to give an overview over important existing approaches to recognize human affect out of the audio and video signal.

4.4.1 Background

Within this section we motivate emotion recognition and the modalities chosen in this article. Also, we introduce models of emotion and talk about databases.

4.4.1.1 Application Scenarios

Even though button pressing starts to be substituted by more natural communication forms such as talking and gesturing, human-computer communication still feels somehow impersonal, insensitive, and mechanical. If we take a comparative glance at human-human communication we will realize a lack of the extra information sensed by man concerning the affective state of the counterpart. This emotional information highly influences the explicit information, as recognized by today's human-machine communication systems, and with an increasingly natural communication, respect of it will be expected. Throughout the design of next generation man-machine interfaces inclusion of this implicit channel therefore seems obligatory [7].

Automatic emotion recognition is nowadays already introduced experimentally in call centers, where an annoyed customer is handed over from a robot to a human call operator [8, 22, 36], and in first commercial lifestyle products as fun software

intended to detect lies, stress- or love level of a telephoner. Besides these, more general fields of application are an improved *comprehension of a user intention*, *emotional accommodation* in the communication (e.g. adaptation of acoustic parameters for speech synthesis if a user seems sad), *behavioral observation* (e.g. whether an airplane passenger seems aggressive) , *objective emotional measurement* (e.g. as a guide-line for therapists), *transmission of emotion* (e.g. sending laughing or crying images within text-based emails), *affect-related multimedia retrieval* (e.g. highlight spotting in a sports event), and *affect-sensitive lifestyle products* (e.g. a trembling cross-hair in video games, if the player seems nervous) [2, 7, 35].

4.4.1.2 Modalities

Human emotion is basically observable within a number of different modalities. First attempts to automatic recognition applied invasive measurement of e.g. the *skin conductivity*, *heart rate*, or *temperature* [37]. While exploitation of this information source provides a reliable estimation of the underlying affect, it is often felt uncomfortable and unnatural, as a user needs to be wired or at least has to stay in touch with a sensor. Modern emotion recognition systems therefore focus rather on video or audio based non-invasive approaches in the style of human emotion recognition: It is claimed that we communicate by 55% visually, through body language, by 38% through the tone of our voice and by 7% through the actual spoken words [29]. In this respect the most promising approach clearly seems to be a combination of these sources. However, in some systems and situations only one may be available.

Interestingly, contrary to most other modalities, speech allows the user to control the amount of emotion shown, which may play an important role, if the user feels too much observed otherwise. Speech-based emotion recognition in general provides reasonable results already by now. However, it seems sure that the visual information helps to enable a more robust estimation [35]. Seen from an economical point of view a microphone as sensor is standard hardware in many HCI-systems today, and also more and more cameras emerge as in cellular phones of today's generation. In these respects we want to give an insight into acoustic, linguistic and vision-based information analysis in search of affect within this chapter, and provide solutions to a fusion of these.

4.4.1.3 Emotion Model

Prior to recognizing emotion one needs to establish an underlying emotion model. In order to obtain a robust recognition performance it seems reasonable to limit the complexity of the model, e.g. kind and number of emotion labels used, in view of the target application. This may be one of the reasons that no consensus exists about such a model in technical approaches, yet. Two generally different views dominate the scene: on the one hand an emotion sphere is spanned by two up to three orthogonal axes: firstly *arousal* or *activation*, respecting the readiness to take some action, secondly *valence* or *evaluation*, considering a positive or negative attitude, and finally *control* or *power*, analyzing the speaker's dominance or submission [7]. While this approach provides a good basis for emotional synthesis, it is often too complex for concrete application scenarios. The better known way therefore is to classify

emotion by a limited set of discrete emotion tags. A first standard set of such labels exists within the MPEG-4 standard comprising *anger, disgust, fear, joy, sadness* and *surprise* [31]. In order to discriminate a non-emotional state it is often supplemented by *neutrality*. While this model opposes many psychological approaches, it provides a feasible basis in technical view. However, further emotions as *boredom* are often used.

4.4.1.4 Emotional Databases

In order to train and test intended recognition engines, a database of emotional samples is needed. Such a corpus should provide spontaneous and realistic emotional behavior out of the field. The sample quality should ensure studio audio and video quality, but for analysis of robustness in the noise also samples with known background noise conditions may be desired. A database further has to consist of a high number of ideally equally distributed samples for each emotion, both of the same, and of many different persons in total. These persons should provide a flashy model considering genders, age groups, ethical backgrounds, among others. Respecting further variability, uttered phrases should possess different contents, lengths, or even languages. Thereby an unambiguous assignment of collected samples to emotion classes is especially hard in this discipline. Also, perception tests by human test-persons are very useful: As we know, it may be hard to rate one's emotion for sure. In this respect it seems obvious that comparatively minor recognition rates can be demanded in this discipline considering related pattern recognition tasks. However, the named human performance provides a reasonable benchmark for a maximum expectation. Finally, a database should be made publicly available in view of international comparability, which seems a problem considering the lacking consensus about emotion classes used and privacy of the test-persons. A number of methods exist to create a database, with arguably different strengths: The predominant ones among these are *acting* or *eliciting* of emotions in test set-ups, hidden or conscious *long-term observations*, and use of clips out of public media content. However, most databases use acted emotions, which allow for fulfillment of the named requirements besides the spontaneity, as there is doubt whether acted emotions are capable of representing true characteristics of affect. Still, they do provide a reasonable starting point, considering that databases of real emotional speech are hard to obtain. In [51] an overview over existing speech databases can be found. Among the most popular ones we want to name the *Danish Emotional Speech Database (CEICES)*, the *Berlin Emotional Speech Database (EMO-DB)*, and the *AIBO Emotional Speech Corpus (AEC)* [3]. Audio-visual databases are however still sparse, especially in view of the named requirements.

4.4.2 Acoustic Information

Basically it can be said that two main information sources are exploited considering emotion recognition from speech: the acoustic information analyzing the prosodic structure as well as the spoken content itself, namely the language information. Hereby the predominant aims besides high reliability are an independence of the

speaker, the spoken language, the spoken content when considering acoustic processing, and the background noise. A number of parameters besides the named underlying emotion model and database size and quality strongly influence the quality in these respects and will be mostly addressed throughout the ongoing: the *signal capturing*, *pre-processing*, *feature selection*, *classification method*, and a reasonable *integration* in the interaction and application context.

4.4.2.1 Feature Extraction

In order to estimate a user's emotion by acoustic information one has to carefully select suited features. Such have to carry information about the transmitted emotion, but they also need to fit the chosen modeling by means of classification algorithms. Feature sets used in existing works differ greatly, but the feature types used in acoustic emotion recognition may be divided into *prosodic* features (e.g. *intensity*, *intonation*, *durations*), *voice quality* features (e.g. 1-7 *formant positions*, 1-7 *formant band widths*, *harmonic-to-noise ratio* (HNR), *spectral features*, 12-15 *Mel Frequency Cepstral Coefficients (MFCC)*), and *articulatory* ones (e.g. *spectral centroid*, and more hard to compute ones as *centralization of vowels*).

In order to calculate these, the speech signal is firstly weighted with a shifting soft window function (e.g. *Hamming window*) of lengths reaching from 10-30ms with a window overlap around 50%. This is a common procedure in speech processing and is needed, as the speech signal is quasi stationary. Next a contour value is computed for every frame and every contour, leading to a multivariate time series. As for intensity mostly simple logarithmic frame energy is computed. However, it should be mentioned that this does not respect human perception. Spectral analysis mostly relies on the Fast Fourier Transform or MFCC - a standard homomorphic spectral transformation in speech processing aiming at de-convolution of the vowel tract transfer function and perceptual modeling by the Mel-frequncy-scale. First problems now arise, as the remaining feature contours pitch, HNR, or formants can only be estimated. Especially pitch and HNR can either be derived out of the spectrum, or - more popular - by peak search within the auto correlation function of the speech signal. Formants may be obtained by analysis of the Linear Prediction Coefficients (LPC), which we will not dig into. Often backtracking by means of dynamic programming is used to ensure smooth feature contours and reduce global costs rather than local ones. Mostly, also higher order derivatives as speed and acceleration are included to better model temporal changes. In any case filtering of the contours leads to a gain by noise reduction, and is done with low-pass filters as moving average or median filters [31, 43].

Next, two in general different approaches exist considering the further acoustic feature processing in view of the succeeding classification: *dynamic* and *static modeling*. Within the dynamic approach the raw feature contours, e.g. the pitch or intensity contours, are directly analyzed frame-wise by methods capable of handling multivariate time-series as dynamic programming (e.g. *Hidden Markov Models (HMM)* [31] or *Dynamic Bayesian Nets (DBN)*).

The second way, by far more popular, is to systematically derive functionals out of the time-series by means of descriptive statistics. Mostly used are thereby *moments*

as mean and standard deviation, or *extrema* and their positions. Also *zero-crossing-rates (ZCR)*, *number of turning points* and others are often considered. As temporal information is thereby mostly lost, duration features are included. Such may be the mean length of pauses or voiced sounds, etc. However, these are more complex to estimate. All features should generally be normalized by either mean subtraction and division by the standard deviance or maximum, as some classifiers are susceptible to different number ranges.

In a direct comparison under constant test conditions the static features outperformed the dynamic approach in our studies [43]. This is highly due to the unsatisfactory independence of the overall contour in respect of the spoken content. In the ongoing we therefore focus on the static approach.

4.4.2.2 Feature Selection

Now that we generated a high order multivariate time-series (approx. 30 dimensions), included delta regression-coefficients (approx. 90 dimensions in total), and started to derive static features in a deterministic way, we end up with a too high dimensionality ($>$300 features) for most classifiers to handle, especially considering typically sparse databases in this field. Also, we would not expect every feature that is generated to actually carry important and non-redundant information about the underlying affect. Still, this costs extraction effort. Likewise, we aim at a dimensionality reduction by *feature selection (FS)* methods.

A often chosen approach thereby is to use *Principal Component Analysis (PCA)* in order to construct superposed-features out of all features, and select the ones with highest eigen-values corresponding to the highest variance [5]. This however does not save the original extraction effort, as still all features are needed for the computation of the artificial ones. A genuine reduction of the original features should therefore be favored, and can be done e.g. by single feature relevance calculation (e.g. *information gain ratio* based on entropy calculation), named *filter-based selection*. Still, the best single features do not necessarily result in the best set. This is why so called *wrapper-based selection* methods usually deliver better overall results at lower dimensionality. The term wrapper alludes to the fact that the target classifier is used as an optimization target function, which helps to not only optimize a set, but rather the compound of features and classifiers as a whole. A search function is thereby needed, as *exhaustive search* is in general NP-hard. Mostly applied among these are *Hill Climbing* search methods (e.g. *Sequential Forward Search (SFS)* or *Sequential Backward Search (SBS)*) which start from a full or empty feature set and step-wisely reduce it by the least relevant or add the most important one. If this is done in a floating manner we have the *Sequential Floating Search Methods (SFSM)* [45]. However, several other mighty such methods exist, among which especially *genetic search* proves powerful [52].

After such reduction the feature vector may be reduced to approximate 100 features, and will be classified in a next step.

4.4.2.3 Classification Methods

A number of factors influence the choice of the classification method. Besides high recognition rates and efficiency, economical aspects and a reasonable integration in the target application framework play a role. In the ongoing research a broad spectrum reaching from rather basic classifiers such as *instance based learners* or *Naive Bayes* to more complex as *Decision Trees*, *Artificial Neuronal Nets* (e.g. *Multi Layer Perceptrons (MLP)* or *Radial Basis Function Networks (RBF)*), and *Support Vector Machines (SVM)* [7, 35]. While the more complex tend to show better results, no general agreement can be found so far. However, SVM tend to be among the most promising ones [42]. The power of such base classifiers can also be boosted or combined my methods of *ensemble construction* (e.g. *Bagging*, *Boosting*, or *Stacking*) [36, 45, 52].

4.4.3 Linguistic Information

Up to here we described a considerable amount of research effort on feature extraction and classification algorithms to the investigation of vocal properties for the purpose of inferring to probably expressed emotions from the sound. So the question after information transmitted within the acoustic channel *"How was it said?"* has been addressed with great success. Recently more attention is paid to the interpretation of the spoken content itself dealing with the related question *"What was said?"* in view of the underlying affect. In psychological studies it is claimed that a connection between certain terms and the related emotion has been learned by the speaker [22]. As hereby the speaker's expression of his emotion consists in usage of certain phrases that are likely to be mixed with meaningful statements in the context of the dialog, an approach with abilities in spotting for emotional relevant information is needed. Consider for this the example *"Could you please tell me much more about this awesome field of research"*. The ratio of affective words is clearly dependent of the underlying application background and the personal nature of the speaker, however it will be mostly very low. It therefore remains questionable whether linguistic information might be sufficient applied standalone. However, its integration showed clear increase in performance [5, 8, 22, 42], even though the conclusions drawn rely per definition on erroneous *Automatic Speech Recognition (ASR)* outputs. In order to reasonably handle the incomplete and uncertain data of the ASR unit, a robust approach should take acoustic confidences into account throughout the processing. Still, none existing system for emotional language interpretation calculates an output data certainty based upon the input data certainty, except for the one presented in [44]. The most trivial approach to linguistic analysis would be the spotting for single emotional terms $w_j \in U$ within an utterance U labeled with an emotion $e_i \in E$ out of the set of emotions E. All known emotional keywords $w_k \in V$ would than be stored within a vocabulary V. In order to handle only emotional keywords, sorting out abstract terms that cannot carry information about the underlying emotion as names helps comparable to the feature space reduction for acoustic features. This is known as *stopping* within linguistics. A so called *stop-list*

can thereby be obtained either by expert-knowledge or by automated approaches as calculation of the salience of a word [22]. One can also cope with emotionally irrelevant information by a normalized log likelihood ratio between an emotion and a general task specific model [8]. Additionally by *stemming* words of the same stem are clustered, which also reduces vocabulary size while in general directly increasing performance. This comes, as hits within an utterance are crucial, and their number increases significantly if none is lost due to minor word differences as plural forms or verb conjunctions.

However, such an approach does not model word order, or the fact, that one term can represent several emotions, which leads us to more sophisticated approaches, as shown in the ongoing.

4.4.3.1 N-Grams

A common approach to speech language modeling is the use of *n-grams*. Let us first assume the conditional probability of a word w_j is given by its predecessors from left to right within an utterance U as $P(w_j|w_1, ..., w_{j-1})$. Next, for language interpretation based emotion recognition *class-based n-grams* are needed. Likewise an emotion e_i within the emotion set E shall have the a-posteriori probability $P(e_i|w_1, ..., w_j)$ given the words w_j and its predecessors in U. However, following *Zipf's principle of least effort*, which states that irrelevant function words occur very frequently, but terms of interest are rather sparse, we reduce the number of considered words to N in order to prevent over-modeling. Applying the *first order Markov assumption* we can therefore use the following estimation:

$$P(e_i|w_1, ..., w_j) \approx P(e_i|w_{j-N-1}, ..., w_j) \tag{4.26}$$

However, mostly uni-grams have been applied so far [8,22], besides bi-grams and trigrams [1], which is due to the very limited typical corpus sizes in speech emotion recognition. Uni-grams provide the probability of an emotion under the condition of single known words, which means they are contained in a vocabulary, without modeling of neighborhood dependencies. Likewise, in a decision process, the actual emotion e can be calculated as:

$$e = \arg_i \max \prod_{(w_j \in U) \wedge (w_j \in V)} P(e_i|w_j) \tag{4.27}$$

Now, in order to calculate $P(e_i|w_j)$, we can use simple *Maximum Likelihood Estimation (MLE)*:

$$P_{MLE}(e_i|w_j) = \frac{TF(w_j, e_i)}{TF(w_j, E)} \tag{4.28}$$

Let thereby $TF(w_j, e_i)$ denote the frequency of occurrence of the term w_j tagged with the emotion e_i within the whole training corpus. $TF(w_j, E)$ then resembles the whole frequency of occurrence of this particular term. One problem thereby is that never occurring term/emotion couples lead to a probability resembling zero. As this is crucial within the overall calculation, we assume that every word has a general

probability to appear under each emotion. This is realized by introduction of the *Lidstone coefficient* λ as shown in the following equation:

$$P_\lambda(e_i|w_j) = \frac{TF(w_j, e_i) + \lambda}{TF(w_j, E) + \lambda \cdot |E|}, \lambda \in [0...1] \tag{4.29}$$

If λ resembles one, this is also known as *Maximum-a-Posteriori (MAP)* estimation.

4.4.3.2 Bag-of-Words

The so called *Bag-of-Words* method is a standard representation form for text in automatic document categorization [19] that can also be applied to recognize emotion [42]. Thereby each word $w_k \in V$ in the vocabulary V adds a dimension to a linguistic vector $\mathbf{x}$ representing the logarithmic term frequency within the actual utterance U known as *logTF* (other calculation methods exist, which we will not show herein). Likewise a component $x_{logTF,j}$ of the vector $\mathbf{x_{logTF}}$ with the dimension $|V|$ can be calculated as:

$$x_{logTF,j} = \log \frac{TF(w_j, U)}{|U|} \tag{4.30}$$

As can be seen in the equation, the term frequency is normalized by the phrase length. Due to the fact that a high dimensionality may decrease the performance of the classifier and flections of terms reduce performance especially within small databases, stopping, stemming, or further methods of feature reduction are mandatory. A classification can now be fulfilled as with the acoustic features. Preferably, one would chose SVM for this task, as they are well known to show high performance [19].

Similar as for the uni-grams word order is not modeled by this approach. However, one advantage is the possibility of direct inclusion of linguistic features within the acoustic feature vector.

4.4.3.3 Phrase Spotting

As mentioned, a key-drawback of the approaches so far is a lacking view of the whole utterance. Consider hereby the negation in the following example: *"I do not feel too good at all,"* where the positively perceived term *"good"* is negated. Therefore *Bayesian Nets (BN)* as a mathematical background for the semantic analysis of spoken utterances may be used taking advantage of their capabilities in spotting and handling uncertain and incomplete information [44]. Thereby each BN consists of a set of N nodes related to state variables X_i, comprising a finite set of states. The nodes are connected by directed edges reaching from *parent* to *child* nodes, and expressing quantitatively the conditional probabilities of nodes and their parent nodes. A complete representation of the network structure and conditional probabilities is provided by the joint probability distribution:

$$P(X_1, ..., X_N) = \prod_{i=1}^{N} P(X_i|parents(X_i)) \tag{4.31}$$

Methods of interfering the states of some query variables based on observations regarding evidence variables are provided by the network. Similar to a standard approach to natural speech interpretation, the aim here is to make the net maximize the probability of the root node modeling the specific emotion expressed by the speaker via his choice of words and phrases. The root probabilities are distributed equally in the initialization phase and resemble the priors of each emotion. If the emotional language information interpretation is used stand-alone, a maximum likelihood decision takes place. Otherwise the root probability for each emotion is fed forward to a higher-level fusion algorithm. On the input layer a standard HMM-based ASR engine with zero-grams as language model providing N-best hypotheses with single word confidences may be applied. In order to deal with the acoustic certainties the traditional BN may be extended to handle soft evidences [44]. The approach discussed here is to be based on integration and abstraction of semantically similar units to higher leveled units in several layers. On the input level the N-best recognized phrases are presented to the algorithm, which maps this input on defined interpretations via its semantic model consisting in a BN.

At the beginning the spotting on items known to the semantic model is achieved by matching the words in the input level to word-nodes contained in the lowest layer of the BN. Within this step the knowledge about uncertainty of the recognized words represented by their confidences is completely transferred into the interpretation model by accordingly setting soft evidence in the corresponding word-nodes. While stepping forward to any superior model-layer, those units resembling each other in their semantic properties regarding the target interpretations are clustered to semantic super-units until the final layer with its root-nodes of the network is reached. Thereby the evidences assigned to word-nodes, due to corresponding appearance in the utterance, finally result in changes of probabilities in the root-nodes representing confidences of each specific emotion and their extent. After all the BN approach allows for an entirely probabilistic processing of uncertain input to gain real probability afflicted output. To illustrate what is understood as semantically similar, consider for instance some words expressing positive attitude, as *"good"*, *"well"*, *"great"*, etc. being integrated into a super-word *"Positive"*. The quantitative contribution $P(e_i|w_j)$ of any word w_j to the belief in an emotion e_i is calculated in a training phase by its frequency of occurrence under the observation of the emotion on basis of the speech corpus as shown within n-grams. Given a word order within a phrase, an important modification of classic BN has to be carried out, as BN's are in general not capable of processing sequences due to their entirely commutative evidence assignment.

4.4.4 Visual Information

For a human spectator the visual appearance of a person provides rich information about his or her emotional state [29]. Thereby several sources can be identified, such as *body-pose* (upright, slouchy), *hand-gestures* (waving about, folded arms), *head-gestures* (nodding, inclining), and - especially in a direct close conversation - the variety of *facial expressions* (smiling, surprised, angry, sad, etc.). Very few ap-

proaches exist towards an affective analysis of body-pose and gestures, while several works report considerable efforts in investigating various methods for facial expression recognition. Therefore we are going to concentrate on the latter in this section, provide background information, necessary pre-processing stages, algorithms for affective face analysis, and give an outlook on forthcoming developments.

4.4.4.1 Prerequisites

Similar to the acoustic analysis of speech for emotion estimation, the task of *Facial Expression Recognition* can be regarded as a common pattern recognition problem with the familiar stages of *preprocessing*, *feature extraction*, and *classification*. The addressed signal, i.e. the camera view to a face, provides lots of information that is neither dependent on nor relevant to the expression. These are mainly ethnic and inter-cultural differences in the way to express feelings, inter-personal differences in the look of the face, gender, age, facial hair, hair cut, glasses, orientation of the face, and direction of gaze. All these influences are quasi disturbing noise with respect to the target source *facial expression* and the aim is to reduce the impact of all noise sources, while preserving the relevant information. This task however constitutes a considerable challenge located in the preprocessing and feature extraction stage, as explicated in the following.

¿From the technical point of view another disturbing source should be minimized - the variation in the position of the face in the camera image. Therefore, the preprocessing comprises a number of required modules for robust head-localization and estimation of the face-orientation. As a matter of fact facial expression recognition algorithms perform best, when the face is localized most accurately with respect to the person's eyes. Thereby, *best* addresses quality and execution time of the method. However, eye localization is a computationally even more complex task than face localization. For this reason a layered approach is proposed, where the search area for eyes is limited to the output-hypotheses of the previous face localizer.

Automatic Facial Expression Recognition has still not arrived in real world scenarios and applications. Existing systems postulate a number of restrictions regarding camera hardware, lighting conditions, facial properties (e.g. glasses, beard), allowed head movements of the person, and view to the face (frontal, profile). Different approaches show different robustness on the mentioned parameters. In the following we want to give an insight in different basic ideas that are applied to automatic mimic analysis. Many of them were derived from the task of face recognition, which is the visual identification or authentication of a person based on his or her face. Hereby, the applied methods can generally be categorized in *holistic* and *non-holistic* or *analytic* [34].

4.4.4.2 Holistic Approaches

Holistic methods (Greek: holon = the whole, all parts together) strive to process the entire face as it is without any incorporated expert-knowledge, like geometric properties or special regions of interest for mimic analysis.

One approach is to extract a comprehensive and respectively large set of features from the luminance representation or textures of the face image. Exemplarily, *Gabor-*

Wavelet coefficients proved to be an adequate parametrization for textures and edges in images [48]. The response of the Gabor filter can be written as a correlation of the input image $I(x)$, with the Gabor kernel $p_k(x)$

$$a_k(x_0) = \int\int I(x)p_k(x - x_0)dx, \tag{4.32}$$

where the Gabor filter $p_k(x)$ can be formulated as:

$$p_k(x) = \frac{k^2}{\sigma^2} exp(-\frac{k^2}{2\sigma^2}x^2) \left(exp\left(ikx\right) - exp\left(-\frac{\sigma^2}{2}\right)\right), \tag{4.33}$$

while k is the characteristic wave vector. Most works [23] apply 5 spatial frequencies with $k_i = \left(\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16}, \frac{\pi}{32}\right)$ and 8 orientations from 0 to π differing by $\pi/8$, while σ is set to the value of π. Consequently, 40 coefficients are computed for each position of the image. Let $I(x)$ be of height 150 pixels and width 100 pixels, likewise $150 \cdot 100 \cdot 40 = 6 \cdot 10^5$ features are computed.

Subsequently, Machine Learning methods for *feature selection* identify the most relevant features as described, which allow for a best possible discrimination of the addressed classes during training phase. A common algorithm applied for this problem was proposed by Freud and Schapire and is known as *AdaBoost.M1* [11, 17]. AdaBoost and its derivates are capable to perform on feature vectors of six-digit dimensionality, while execution time remains tolerable. On the way to assignment of the reduced static feature set to emotional classes, any statistical algorithm can be used. In case of video processing and real-time requirements the choice of classifiers might focus on linear methods or decision trees, depending on the computational effort to localize the face and extract the limited number of features.

In Face Recognition the approach of Eigenfaces proposed by Turk and Pentland has been examined thoroughly [49]. Thereby, the aim is to find the principal components of the distribution of two-dimensional face representations. This is achieved by the determination and selection of Eigen-vectors from the covariance matrix of a set of representative face images. This set should cover different races for the computation of the covariance matrix. Each image of size $N \times M$ pixels, which can be thought of as a $N \times M$ matrix of 8bit luminance values, is transformed into a vector of dimensionality $N \cdot M$. Images of faces, being similar in overall configuration, are not randomly distributed in this very high-dimensional image space, and thus can be described by a relatively low dimensional subspace, spanned by the relevant Eigenvectors. This relevance is measured by their corresponding Eigen-values, indicating a different amount of variation among the faces. Each image pixel contributes more or less to each Eigen-vector, so that each of them can be displayed, resulting in a kind of ghostly face - the so called Eigenface. Finally, each individual face can be represented exactly by linear combination of these Eigenfaces, with a remaining error due to the reduced dimensionality of the new face space. The coefficients or weights of that linear combination that minimize the error between the original image and the face space representation now serve as features for any kind of classification. In case of Facial Expression Recognition the covariance matrix would be computed on a

set of faces expressing all categories of mimics. Subsequently, we apply the Eigenvector analysis and extract representative sets of weights for each mimic-class that should be addressed. During classification of unknown faces, the weights for this image are computed accordingly, and will then be compared to weight-vectors of the training set.

4.4.4.3 Analytic Approaches

These methods concentrate on the analysis of dominant regions of interest. Thus, pre-existing knowledge about geometry and facial movements is incorporated, so that subsequent statistical methods benefit [9]. Ekman and Friesen introduced the so called Facial Action Coding System in 1978, which consists of 64 *Action Units (AU)*. Presuming that all facial muscles are relaxed in the neutral state, each AU models the contraction of a certain set of them, leading to deformations in the face. Thereby the focus lies on the predominant facial features, such as *eyes*, *eye-brows*, *nose*, and *mouth*. Their shape and appearance contain most information about the facial expressions. One approach for analyzing shape and appearance of facial features is known as *Point Distribution Model (PDM)*. Cootes and Taylor gave a comprehensive introduction to the theory and implementation aspects of PDM. Subclasses of PDM are *Active Shape Models (ASM)* and *Active Appearance Models (AAM)*, which showed their applicability to mimic analysis [18].

Shape models are statistical descriptions of two-dimensional relations between landmarks, positioned on dominant edges in face images. These relations are freed of all transformations, like rotation, scaling and translation. The different shapes that occur just due to the various inter-personal proportions are modeled by PCA of the observed landmark displacements during training phase. The search of the landmarks starts with an initial estimation where the shape is placed over a face manually or automatically, when preprocessing stages allow for. During search, the edges are approximated by an iterative approach that tries to measure the similarity of the sum of edges under the shape to the model. Appearance Models additionally investigate the textures or gray-value distributions over the face, and combine this knowledge with shape statistics.

As mentioned before, works in the research community try to investigate a broad range of approaches and combinations of different holistic and analytic methods in order to proceed towards algorithms, that are robust to the broad range of different persons and real-life situations. One of the major problems is still located in the immense computational effort and applications will possibly have to be distributed on multiple CPU and include the power of GPU to converge to real-time abilities.

4.4.5 Information Fusion

In this chapter we aim to fuse the acoustic, linguistic and vision information obtained. This integration (see also Sect. 4.3.3 is often done in a *late semantic manner* as *(weighted) majority voting* [22]. More elegant however is the direct fusion of the streams within one feature vector [42], known as the *early feature fusion*. The advantage thereby is that less knowledge is lost prior to the final decision. A compromise

between these two is the so called *soft decision fusion* whereby the confidence of each classifier is also respected. Also the integration of a N-best list of each classifier is possible. A problem however is the synchronization of video, and the acoustic and linguistic audio streams. Especially if audio is classified on a global word or utterance level, it may become difficult to find video segments that correspond to these units. Likewise, audio processing may be preferred by dynamic means in view of early fusion with a video stream.

4.4.6 Discussion

Especially automatic speech emotion recognition based on acoustic features already comes close to human performance [42] somewhere around 80% recognition performance. However, usually very idealized conditions as known speakers and studio recording conditions are considered, yet. Video processing does not reach these regions at the time, and conditions are very ideal, as well. When using emotion recognition systems out-of-the-lab, a number of new challenges arise, which has been hardly addressed within the research community up to now. In this respect the future research efforts will have to lead to larger databases of spontaneous emotions, robustness under noisy conditions, less person-dependency, reliable confidence measurements,integration of further multimodal sources, contextual knowledge integration, and acceptance studies of emotion recognition applied in everyday systems. In this respect we are looking forward to a flourishing human-like man-machine communication supplemented by emotion for utmost naturalness.

References

1. Ang, J., Dhillon, R., Krupski, A., Shriberg, E., and Stolcke, A. Prosody-Based Automatic Detection of Annoyance and Frustration inHuman-Computer Dialog. In *Proceedings of the International Conference on Speech and LanguageProcessing (ICSLP 2002)*. Denver, CO, 2002.
2. Arsic, D., Wallhoff, F., Schuller, B., and Rigoll, G. Video Based Online Behavior Detection Using Probabilistic Multi-StreamFusion. In *Proceedings of the International IEEE Conference on Image Processing(ICIP 2005)*. 2005.
3. Batliner, A., Hacker, C., Steidl, S., Nöth, E., Russel, S. D. M., and Wong, M. 'You stupid tin box' - Children interacting with the AIBO robot:A cross-linguisitc emotional speech corpus. In *Proceedings of the LREC 2004*. Lisboa, Portugal, 2004.
4. Benoit, C., Martin, J.-C., Pelachaud, C., and andB. Suhm, L. S., editors. *Audiovisual and Multimodal Speech Systems.* In: Handbook of Standards and Resources for Spoken Language Systems- SupplementVolume.D. Gibbon, I. Mertins, R.K. Moore , Kluwer InternationalSeriesin Engineering and Computer Science, 2000.
5. Chuang, Z. and Wu, C. Emotion Recognition using Acoustic Features and Textual Content. In *Proceedings of the International IEEE Conference on Multimedia andExpo (ICME) 2004*. Taipei, Taiwan, 2004.
6. Core, M. G. Analyzing and Predicting Patterns of DAMSL Utterance Tags. In *AAAI Spring Symposium Technical Report SS-98-01*. AAAI Press, 1998. ISBN ISBN 1-57735-046-4.

7. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., andS. Kollias, G. V., Fellenz, W., and Taylor, J. G. Emotion recognition in human-computer interaction. *IEEE Signal Processing magazine*, 18(1):32–80, January 2001.
8. Devillers, L. and Lamel, L. Emotion Detection in Task-Oriented Dialogs. In *Proceedings of the International Conference on Multimedia and Expo(ICME 2003), IEEE, Multimedia Human-Machine Interface and InteractionI*, volume III, pages 549–552. Balitmore, MD, 2003.
9. Ekman, P. and Friesen, W. Facial Action Coding System. *Consulting Psychologists Press*, 1978.
10. et al., S. O. Designing the user interface for multimodal speech and pen-basedgestureapplications: State-of-the-art systems and future directions. *Huma Computer Interaction*, (15(4)):263–322, 2000.
11. Freund, Y. and Schapire, R. Experiments with a New Boosting Algorithm. In *International Conference on Machine Learning*, pages 148–156. 1996.
12. Geiser, G., editor. *Mensch-Maschine-Kommunikation*. Oldenbourg-Verlag, München, 1990.
13. Goldschen, A. and Loehr, D. The Role of the DARPA Communicator Architecture as a Human-ComputerInterface for Distributed Simulations. In *Simulation Interoperability Standards Organization (SISO) SpringSimulation Interoperability Workshop*. Orlando, Florida, 1999.
14. Grosz, B. and Sidner, C. Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.
15. Hartung, K., Münch, S., and et al, L. S. MIAMI: Software Architecture, Deliverable Report 4. Report of ESPRITIII:Basic Research Project 8579, Multimodal Interface for AdvancedMultimediaInterfaces (MIAMI). Technical report, 1996.
16. Hewett, T., Baecker, R., Card, S., Carey, T., andM. Manteiand G. Perlman, J. G., Strong, G., and Verplank, W., editors. *Curricula for Human-Computer Interaction*. ACM Special Interest Group on Computer-Human Interaction, CurriculumDevelopmentGroup, 1996.
17. Hoch, S., Althoff, F., McGlaun, G., and Rigoll, G. BIMODAL FUSION OF EMOTIONAL DATA IN AN AUTOMOTIVE ENVIRONMENT. In *Proc. of the ICASSP 2005, IEEE Int. Conf. on Acoustics, Speech, andSignal Processing*. 2005.
18. Jiao, F., Li, S., Shum, H., and Schuurmanns, D. Face Alignment Using Statistical Models and Wavelet Features. In *Conference on Computer Vision and Pattern Recognition*. 2003.
19. Joachims, T. Text Categorization with Support Vector Machines: Learning with ManyRelevant Features. Technical report, LS-8 Report 23, Dortmund, Germany, 1997.
20. Krahmer, E. The Science and Art of Voice Interfaces. Technical report, Philips Research, Eindhoven, Netherlands, 2001.
21. Langley, P., Thompson, C., Elio, R., and Haddadi, A. An Adaptive Conversational Interface for Destination Advice. In *Proceedings of the Third International Workshop on Cooperative InformationAgents*. Springer, Uppsala, Sweden, 1999.
22. Lee, C. M. and Pieraccini, R. Combining acoustic and language information for emotion recognition. In *Proceedings of the International Conference on Speech and Language-Processing (ICSLP 2002)*. Denver, CO, 2002.
23. Lee, T. S. Image Representation Using 2D Gabor Wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.
24. Levin, E., Pieraccini, R., and Eckert, W. A stochastic Model of Human-Machine Interaction for Learning DialogStrategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23, 2000.

25. Litman, D., Kearns, M., Singh, S., and Walker, M. Automatic Optimization of Dialogue Management. In *Proceedings of the 18th International Conference on ComputationalLinguistics*. Saarbrücken, Germany, 2000.
26. Maybury, M. T. and Stock, O., editors. *Multimedia Communication, including Text.* In: E. Hovy and N. Ide and R. Frederking and J. Mariani and A. Zampolli:MultilingualInformation Management: Current Levels and Future Abilities.A study-commissioned by the US National Science Foundation and also deliveredtoEuropean Commission Language Engineering Office and the US DefenseAdvancedResearch Projects Agency,, 1999.
27. McGlaun, G., Althoff, F., Lang, M., , and Rigoll:, G. Development of a Generic Multimodal Framework for Handling ErrorPatternsduring Human-Machine Interaction. In *SCI 2004, 8th World Multi-Conference on Systems, Cybernetics, andInformatics,Orlando, FL, USA*. 2004.
28. McTear, M. F. *Spoken dialogue technology: toward the conversational user interface*. Springer Verlag, London, 2004. ISBN 1-85233-672-2.
29. Mehrabian, A. Communication without words. *Psychology Today*.
30. Nielsen, J. *Usability Engineering*. Academic Press, Inc., 1993. ISBN 0-12-518405-0.
31. Nogueiras, A., Moreno, A., Bonafonte, A., and Marino, J. Speech Emotion Recognition Using Hidden Markov Models. In *Eurospeech 2001 Poster Proceedings*, pages 2679–2682. Scandinavia, 2001.
32. Oviatt, S. Ten Myths of Multimodal Interaction. *In: Communications of the ACM 42*, 11:74–81, 1999.
33. Oviatt, S., Caulston, R., and Lunsford, R., editors. *When Do We Interact Multimodally? Cognitive Load and Multimodal CommunicationPatterns.* 2004.
34. Pantic, M. and Rothkrantz, L. Automatic Analysis of Facial Expressions: The State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.
35. Pantic, M. and Rothkrantz, L. Toward an Affect-Sensitive Multimodal Human-Computer Interaction. *Proccedings of the IEEE*, 91:1370–1390, September 2003.
36. Petrushin, V. Emotion in Speech: Recognition and Application to Call Centers. In *Proceedings of the Conference on Artificial Neural Networks in Engineering(ANNIE '99)*. 1999.
37. Picard, R. W. *Affective Computing*. MIT Press, Massachusetts, 2nd edition, 1998. ISBN 0-262-16170-2.
38. Pieraccini, R., Levin, E., and Eckert, W. AMICA: The AT&T mixed initiative conversational architecture. In *Proceedings of the Eurospeech '97*, pages 1875–1878. Rhodes, Greece, 1997.
39. Reason, J. *Human Error*. Cambridge University Press, 1990. ISBN 0521314194.
40. Sadek, D. and de Mori, R. Dialogue Systems. In de Mori, R., editor, *Spoken Dialogues with computers*, pages 523–562. Academic Press, 1998.
41. Schomaker, L., Nijtmanns, J., Camurri, C., Morasso, P., and al.:, C. A Taxonomy of multimodal interaction in the human information processingsystem.Report of ESPRIT III: Basic Research Project 8579, Multimodal InterfaceforAdvanced Multimedia Interfaces (MIAMI). Technical report, 1995.
42. Schuller, B., Müller, R., Lang, M., and Rigoll, G. Speaker Independent Emotion Recognition by Early Fusion of Acousticand Linguistic Features within Ensembles. In *Proceedings of the ISCA Interspeech 2005*. Lisboa, Portugal, 2005.
43. Schuller, B., Rigoll, G., and Lang, M. Hidden Markov Model-Based Speech Emotion Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech,and Signal Processing (ICASSP 2003)*, volume II, pages 1–4. 2003.

44. Schuller, B., Rigoll, G., and Lang, M. Speech Emotion Recognition Combining Acoustic Features and LinguisticInformation in a Hybrid Support Vector Machine - Belief NetworkArchitecture. In *Proceedings of the IEEE International Conference on Acoustics, Speech,and Signal Processing (ICASSP 2004)*, volume I, pages 577–580. Montreal, Quebec, 2004.
45. Schuller, B., Villar, R. J., Rigoll, G., and Lang, M. Meta-Classifiers in Acoustic and Linguistic Feature Fusion-BasedAffect Recognition. In *Proceedings of the International Conference on Acoustics, Speechand Signal Processing (ICASSP) 2005*, volume 1, pages 325–329. Philadelphia, Pennsylvania, 2005.
46. Shneiderman, B. *Designing the user interface: Strategies for effective human-computer interaction (3rd ed.)*. Addison-Wesley Publishing, 1998. ISBN 0201694972.
47. Smith, W. and Hipp, D. *Spoken natural language dialog systems: A practical approach*. Oxford University Press, 1994. ISBN 0-19-509187-6.
48. Tian, Y., Kanade, T., and Cohn, J. Evaluation of Gabor-wavelet-based facial action unit recognitionin image sequences of increasing complexity. In *Proceedings of the Fifth IEEE International Conference on AutomaticFace and Gesture Recognition*, pages 229–234. May 2002.
49. Turk, M. and Pentland, A. Face Recognition Using Eigenfaces. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 586–591. 1991.
50. van Zanten, G. V. User-modeling in Adaptive Dialogue Management. In *Proceedings of the Eurospeech '99*, pages 1183–1186. Budapest, Hungary, 1999.
51. Ververidis, D. and Kotropoulos, C. A State of the Art Review on Emotional Speech Databases. In *Proceedings of the 1st Richmedia Conference*, pages 109–119. Lausanne, Sitzerland, 2003.
52. Witten, I. H. and Frank, E. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, CA, 2000. ISBN 1-558-60552-5.
53. Wu, L., Oviatt, S., and Cohen, P. Multimodal integration - A statistical review. *IEEE Transactions on Multimedia*, (1(4)):334–341, 1999.
54. Young, S. Probabilistic Methods in Spoken Dialogue Systems. *Philosophical Transactions of the Royal Society*, 358:1389–1402, 2000.