IN DEFENSE OF A PROBABILISTIC VIEW ON HUMAN WORD RECOGNITION

Dick R. van Bergem

1. INTRODUCTION

Human listeners have an amazing ability to recognize words in fluent speech. They can perform this task very fast and very efficiently, despite the fact that the speech they hear is continuous, highly variable, and often accompanied by some kind of noise. Several models of word recognition have been proposed such as the Logogen model (Morton, 1982), the Search model (Forster, 1979), the Cohort model (Marslen-Wilson & Welsh, 1978; Marslen-Wilson, 1987), TRACE (McClelland & Elman, 1986), and LAFS (Klatt, 1989).

Recently a lot of progress has been made in the field of automatic speech recognition. Generally speaking, two competing strategies are used: knowledge-based strategies and strategies based on stochastic models. In knowledge-based approaches an attempt is made to integrate all our knowledge of speech in a recognition system (e.g. Zue, 1985; Cole et al., 1986). Stochastic models such as neural networks (e.g. McClelland & Elman, 1986) and hidden Markov models (e.g. Jelinek, 1985; Lee, 1989) use large amounts of training data to build up their own internal representation of speech units.

Currently the stochastic models, in particular hidden Markov models, are far more successful in recognizing speech than the knowledge-based models. Nevertheless, some researchers (e.g. Fant, 1990) claim that the 'brute-force' stochastic models will be outperformed by models that are based on our detailed knowledge of speech, as soon as these knowledge-based models have been properly implemented. That is, they claim that our 'real' knowledge is more powerful than the stochastic knowledge of machines. However, one may wonder whether the knowledge gathered by a stochastic model from a large amount of training data is really that different from the knowledge we claim to have about speech.

In the first place stochastic models are very well capable of extracting features from acoustic input. This was shown by Elman and Zipser (1988) who used a neural network to recognize the spectral pattern of the consonant and vowel part of the syllables [ba], [bi], [bu], [da], [di], [du], [ga], [gi], [gu]. For this purpose they chose the backpropagation algorithm (Rumelhart et al., 1986) to train a neural net with one hidden layer. After training they examined the response patterns of the hidden units and found that these units had extracted some phonological features from the training data. One hidden unit was for instance always 'on' for the vowel parts and 'off' for the consonant parts; another hidden unit was sensitive to alveolar stops. Although only a small part of the response patterns of the hidden units could be explained in terms of phonological features, it is beyond doubt that the neural nets use *some* kind of features to represent the acoustic data. It is not unthinkable that these features which are based on a stochastic optimality principle turn out to be more realistic than the phonological features based on 'intuitions' of phoneticians and linguists.

In the second place our human knowledge of speech is also often of a stochastic nature. We know for instance that vowels in stressed syllables are longer than vowels in unstressed syllables (see e.g. Van Bergem, 1990). But where does this 'knowledge' stem from? We measure the duration of a lot of vowels both in stressed and in

unstressed syllables, calculate means and standard deviations for both groups and compare these measures. Is this not in fact the stochastic modelling of a large set of 'training data'?
One might oppose that in this case the researcher has learnt that the division of vowels into a stressed group and an unstressed group makes sense. This is true, but does the notion of stress actually contribute to a better recognition strategy? Stress is not a concrete property of the acoustic signal, but only an abstract linguistic concept that comes forward from *reflection* on the acoustic structure of words in our language. Consequently, stress cannot be directly extracted from the acoustic signal, but it should be detected through its acoustic correlates such as duration, energy, or pitch movements. Apart from the fact that the detection of stress in this way cannot be done perfectly, but only in a probabilistic way (!), it seems much easier to use the acoustic correlates of stress themselves in the recognition process. This is exactly what is done in the stochastic models. Consider for instance two stochastic models for the words conTENT and CONtent (stress indicated by capitals). Training data will cause, among other things, a difference in modelled duration between the stressed /ɛ/ and the unstressed /ɛ/. In this way linguistic concepts such as stress are *implicitly* implemented in stochastic models.
Recently, Elman (1989) has discussed the importance of connectionist approaches to human speech processing. In this article we will discuss the importance of the Markov approach to human speech processing. The artificial neural networks that are used in the connectionist approach are clearly inspired on the neural properties of human memory (In section 3.1 this memory organization is discussed). This is not the case with Markov models which have a purely mathematical basis. Our justification for discussing Markov models in relation to human speech processing is that they are very well suited to model speech and that the *strategies* by which they do this, may be very similar to the ones human listeners use. The discussion will be based on four important problems in the field of human word recognition:

1. What kind of intermediate units (e.g. phonemes, syllables) are used by listeners to get from the acoustic input to some kind of abstract representation of words?
2. How do listeners cope with the enormous amount of variability in the sounds they hear?
3. How are words isolated by listeners from the continuous flow of speech?
4. In what way do listeners use higher level knowledge (e.g. syntactics, semantics) to efficiently extract words from the acoustic flow of speech?

Most word recognition models in the psycholinguistic field focus on one or two of these problems and ignore the others or only vaguely refer to them. In this article all four questions will be discussed in section 3.1 to 3.4 based on the Markov approach in speech recognition. It is not our aim to contrast the psycholinguistic models with Markov models. We merely want to demonstrate the efficiency and plausibility of the probabilistic approach in speech recognition. Before we address the topics mentioned above, we will briefly discuss the basic principles of Markov models.

2. MARKOV MODELS

A Markov model is a probabilistic model that is composed of a number of states and transition probabilities between these states. In figure 1 a simple example of a (fictitious) Markov model is given that 'predicts' tomorrow's weather on the basis of today's weather in Holland (We restrict ourselves to two types of weather). This model has two states: S_1 (Rain) and S_2 (Sun). If we have rain today, the probability of rainy

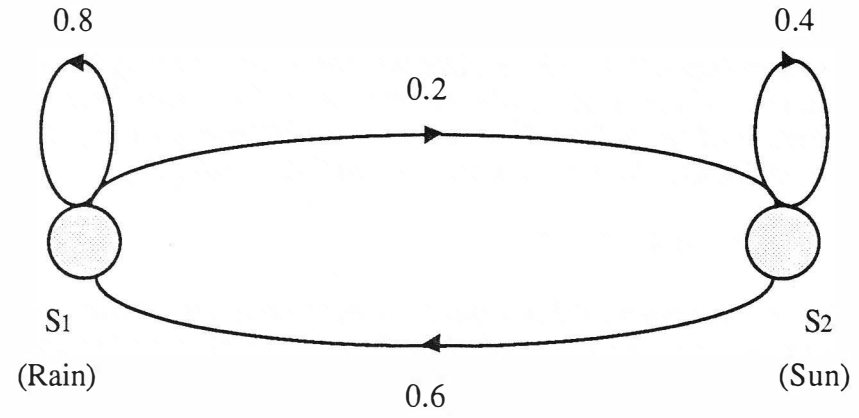


Figure 1. A Markov model that 'predicts' the weather in Holland.

weather tomorrow is 0.8 (80%) and of sunny weather 0.2 (20%). If on the other hand we have sun today, the probability of sunny weather tomorrow is 0.4 (40%) and of rainy weather 0.6 (60%).
It should be noted that *time* plays an important role in Markov models: We move from one state to another in successive time steps (in our example days), so in fact we are modelling the course of events. The self-loops in the model provide the possibility to stay for a number of time steps (days) in the same state. In our example it is more likely to stay in the 'rain'-state than in the 'sun'-state, so it is more likely to have a number of rainy days in succession than a number of sunny days.
Another important aspect of Markov models is the *training*. How do we obtain the transition probabilities? In order to train our weather model we have to observe the weather for a long period and simply count the occurrences of all possible successions of weather types from each day to the next: R_i - R_{i+1}, R_i - S_{i+1}, S_i - S_{i+1}, S_i - R_{i+1} (R = rain, S = sun, i = day i). Suppose we have observed the weather for 1000 days and have found the following counts:

	R_{i+1}	S_{i+1}
R_i	800	200
S_i	600	400

These counts give rise to the transition probabilities shown in figure 1. Obviously, the reliability of the model increases with a larger amount of training data.

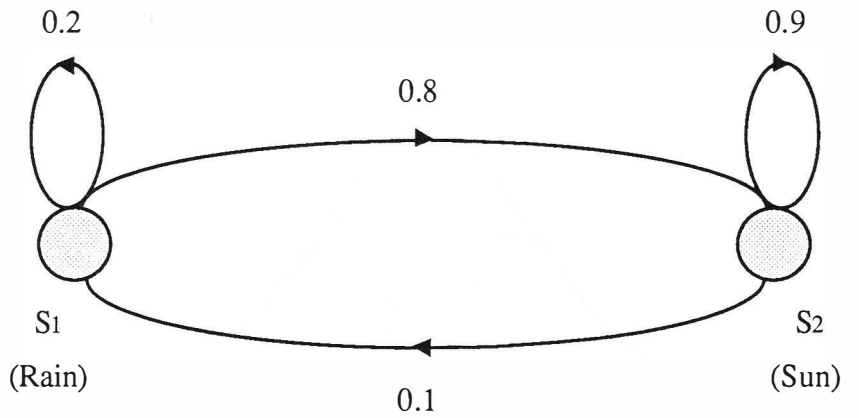


Figure 2. A Markov model that 'predicts' the weather in Italy.

Suppose we have also trained a similar Markov model for the weather in Italy. This model is shown in figure 2. It will be clear that the model for Italy has a 'preference' for long sequences of sunny days, whereas the model for Holland has a 'preference' for long sequences of rainy days. Now consider the following observed sequence of weather types (called the *observation sequence*) for 7 successive days:

$$O_{1..7} = R\ S\ S\ R\ R\ S\ S$$

We may ask ourselves whether this sequence was observed in Holland or in Italy. To answer this question we can simply calculate the total probability of this sequence (the *output probability*) for each model by multiplying the successive transition probabilities. For the Dutch model we find an output probability of:

$$P_D = 0.2 \times 0.4 \times 0.6 \times 0.8 \times 0.2 \times 0.4 = 0.003072$$

and for the concurring Italian model we find an output probability of:

$$P_I = 0.8 \times 0.9 \times 0.1 \times 0.2 \times 0.8 \times 0.9 = 0.010368$$

Since P_I is larger than P_D this weather sequence was more likely observed in Italy. If the models are properly trained, we can use them in this way to *recognize* an observation sequence.
In principle there are two ways of recognizing the input (i.e. an observation sequence). The first one is to compare the output probabilities of all the concurring models and choose the model with the highest probability (or alternatively make a hierarchy of more and less probable models). In this case the absolute value of the probabilities plays no role. This is the strategy that is applied in speech recognition with Markov models. Notice that in this case the input is always recognized. The second way is to compare the output probability of each model separately with the *average* output probability of that model. That is, a sample distribution of output probabilities could be made, if a large sample of correct input strings was fed to a trained model (The output probabilities would have to be normalized for input strings of different lengths). As shown in figure 3, the recognition decision could now be based on a critical value P_c in this distribution. Recognition would occur for input that would generate an output probability exceeding this threshold. Notice that this strategy, which resembles the 'activation level' concept in the Logogen model, might lead to recognition of the input by several models simultaneously or to no recognition at all. The output probability sample distribution could be very useful in evaluating the plausibility of the model for the given input. In this way it could be used to properly deal with 'nonsense' input. This point will be further discussed in section 3.2.

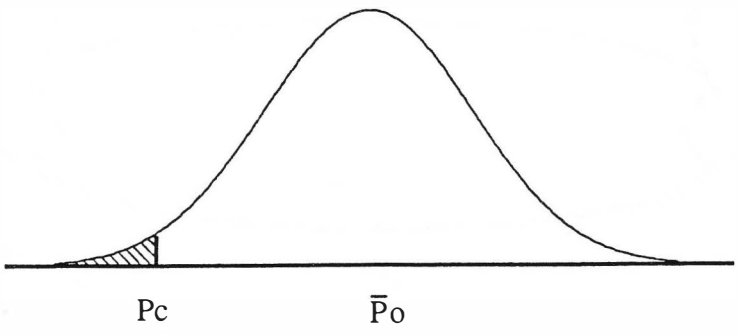


Figure 3. Sample distribution of output probabilities for a single Markov model.

In automatic speech recognition the more powerful *hidden* Markov models are used. Any speech unit can be modelled e.g. phones, phonemes, diphones, syllables, or words. In the following we will not go into the mathematical background of hidden Markov models (interested readers are referred to e.g. Van Alphen and Van Bergem, 1989), but we will focus exclusively on the recognition strategies that are used. For this purpose we will present things somewhat simpler than they really are, without violating the basic ideas.

In 'normal' Markov models the observations are directly associated with states. This is no longer the case in hidden Markov models, where for each state a separate layer is constructed with *observation probabilities*. The observations are usually some kind of spectral representation of the acoustic signal (e.g. LPC-coefficients or bandfilter-values) measured at discrete time steps (typically 10 ms). The entire (high-dimensional) spectral space is split up into a number of non-overlapping areas with a technique called vector quantization and the spectral representation of each occurring sound can be mapped on to one of these areas. When a particular sound is uttered repeatedly, there will usually be significant spectral differences between individual occurrences of the sound and consequently some of the uttered sounds may be mapped on to different spectral areas than others. The observation probability distribution is obtained by counting the number of times the occurrences of a particular sound (in a spectral representation) are mapped on to each of the spectral areas.

To illustrate these ideas, consider two hidden Markov models for the words conTENT and CONtent. For the sake of simplicity, it is easiest to imagine that each phoneme of the words is modelled by just one state (In reality the number of states can be freely chosen and the acoustic evidence is divided in a statistically optimal sense over the states). The durations of the stressed /ɛ/ and the unstressed /ɛ/ are modelled by different self-loop probabilities in the /ɛ/-state. The state associated with the stressed /ɛ/ has a higher self-loop probability which means that the model 'prefers' to stay in that state for a longer stretch of time. Apart from an effect on duration, stress may also effect the spectral quality of the vowels (see e.g. Van Bergem, 1990). The occurrences of the stressed /ɛ/ will be 'full' vowels most of the time, whereas the occurrences of the unstressed /ɛ/ may show a greater variation in spectral qualities ranging from a 'full' vowel to a more or less schwa-like sound. This will result in a peaked observation probability distribution for the stressed /ɛ/ and a flat observation probability distribution for the unstressed /ɛ/. The (fictitious) distributions for the stressed /ɛ/ and the unstressed /ɛ/ are given in figure 4. In this (fictitious) example 250 spectral areas were defined, each of which is simply indexed by an integer number between 1 and 250 (It is assumed that consecutive integers indicate adjacent spectral areas). None of the spectral areas has a probability zero (although some have an extremely small probability),

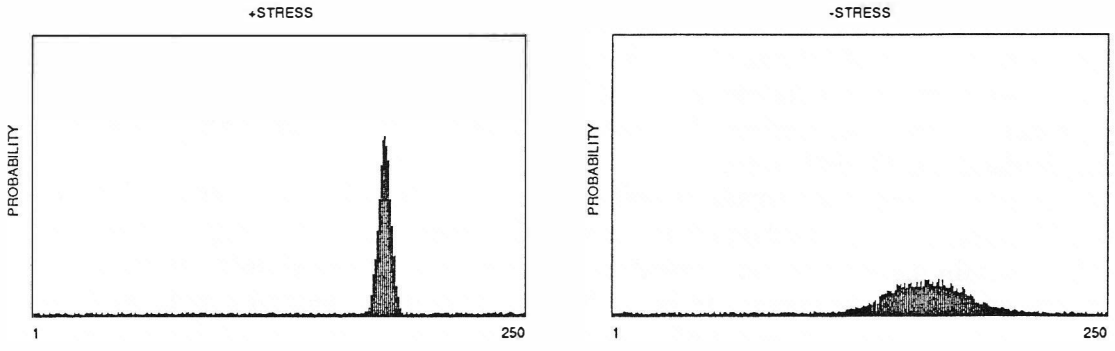


Figure 4. The (fictitious) observation probability distributions for the stressed /ɛ/ of the word conTENT (on the left) and the unstressed /ɛ/ of the word CONtent (on the right).

because in principle any area may match the vowel /ɛ/ in an infinite amount of training data. Of course the other parts of the words conTENT and CONtent are modelled in a similar way. Apart from spectral quality, also energy or pitch can be taken into account. The output probability of the model is obtained by combining (multiplying) the transition probabilities and observation probabilities of all states. Stressed vowels are modelled with high transition (self-loop) probabilities and high observation probabilities and so they can significantly raise the output probability of the correct word model with respect to the other models. In general, the reliable parts of a word are in particular responsible for a relatively high output probability of the appropriate word model and therefore the recognition of a word is for the greater part determined by its reliable parts, which is a desirable property.
Another important aspect of hidden Markov models is that the acoustic evidence for a word can be directly matched with the model without intermediate levels of representation. The stressed and unstressed variants of the vowel /ɛ/ in the words conTENT and CONtent for instance are modelled with their own characteristic areas in the spectral space. Explicit rules that account for coarticulation effects, reduction effects, assimilation effects etc. are *not* needed by the model. Instead, knowledge about these effects is *implicitly* present, because the most probable areas in the spectral space are *determined* by these effects.

3. IMPORTANT TOPICS IN HUMAN WORD RECOGNITION

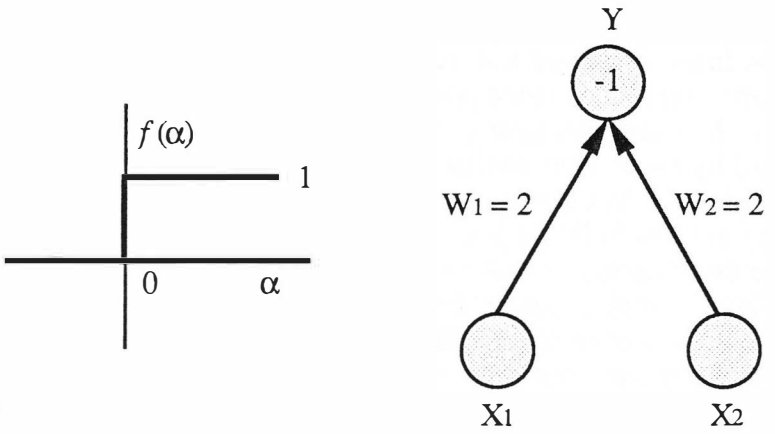
3.1 The intermediate units in word recognition

Most psycholinguistic word recognition models propose a bottom-up flow of the acoustic input through one or more intermediate levels to some kind of abstract representation of words. The units at these intermediate levels are e.g. features, phonemes, or syllables. Klatt (1989) already pointed out, that intermediate representations of the acoustic signal may lead to an accumulation of errors which can severely degrade the recognition performance.
Consider for instance the case that phoneme models are used to build up the words conTENT and CONtent. It will be clear that *labelling* of the second vowel in these words as /ɛ/ does not do justice to the acoustic reality that it can have a variety of spectral qualities (different for the stressed and the unstressed variant), each with a certain probability of occurrence. Moreover, the distinction between conTENT and CONtent will fade away, if only one phoneme model is used for both the stressed and the unstressed /ɛ/. There could of course be two phoneme models for the /ɛ/, one for the stressed variant and one for the unstressed variant. However, in reality an entire lexicon of words has to be modelled, in which the /ɛ/-phoneme occurs each time in a different spectral shape, not only due to stress, but also due to coarticulation, assimilation, etc. And so a lot of different /ɛ/-models would be needed. The problem is how to select a set of /ɛ/-models that is representative of all the existing varieties. The use of *rules* for coarticulation, assimilation, etc. would not be very helpful, because they require the recognition of adjacent phonemes which are also 'coloured' by coarticulation, assimilation, etc.
The choice of larger intermediate units (e.g. syllables) as building blocks for words would introduce less problems than the use of phonemes, because e.g. coarticulatory effects within these units can be modelled. However, the most simple and robust way of recognizing words appears to be an immediate map of spectral events with word templates that were also modelled as strings of spectral events. As we have seen, this can be efficiently done with the approach used in hidden Markov models.

In practice, however, the use of such word models in the Markov approach causes two problems for large lexicons. In the first place it is hard to get enough training data for a proper modelling of the words and in the second place the word models require a lot of (computer) memory space. If a large lexicon of words has to be modelled, these problems are often avoided by using intermediate units of speech after all (usually diphones or triphones), despite the fact that recognition performance decreases. The intermediate units, that are used as building blocks for words, do not need a lot of memory space and they are much more frequently represented in the training data than words. It is interesting to find out, if these two problems would also apply to humans.
The amount of data, that is used to train hidden Markov models, is usually no more than a few hours of speech. More data would be impractical, because the training of hidden Markov models is rather time consuming. The research situation requires that an entire lexicon of words is trained in one session with a relatively small amount of training data to get results as quickly as possible. To meet these demands, suboptimal training methods have to be chosen. For humans this situation is clearly very different. Young babies begin to build up a very restricted lexicon of words and have an abundance of 'training examples' of these words at their disposal. During childhood, the number of words in the lexicon can gradually grow, until a large lexicon has been built up by the time adulthood is reached. In this long period a hugh amount of 'training examples' passes the mind, so that word models can be perfectly tuned.
With regard to the problem of memory capacity, it appears that humans can store an enormous amount of patterns and that they posses a very efficient retrieval mechanism. Elman (1989) pointed out, that we are inclined to think of memory as a kind of 'box-structure' in the way it is used in a (von Neumann) computer. In such a passive memory structure all patterns that are stored will be placed in a separate 'box' and each new pattern thus requires an extra 'box'. This means that the required memory space is directly related to the number of patterns that are stored. Unknown input has to be matched with each of the patterns ('boxes') one at a time and therefore the retrieval time is also directly related to the number of patterns that are stored. To illustrate the way a memory can be differently organized, consider the simple artificial neural network shown in figure 5 that simulates the logical OR-function. It consists of two input nodes X_1 and X_2, one output node Y, weighted connections between them (W_1 and W_2) and a hard limiter $f(\alpha)$ that transforms the output values to a binary code. The output can be calculated as the weighted sum of all input values plus an 'offset' value (which is -1 in this case). The hard limiter shown on the left of figure 5 transforms all positive output values to '1' and all negative output values to '0':

$$Y = f\left(-1 + \sum_{i=1}^{2} W_i X_i\right)$$

The crucial thing to notice is that all four occurring patterns are recognized by *one and the same* network. In addition, the computation of the output value always takes the same time regardless of the number of patterns that are stored. The strategy that is used in a neural network, is to adapt ('train') the connection weights in such a way that all patterns are optimally recognized. Since the pathways from the input nodes to the output nodes form a parallel network of connections, all information conveyed in the input nodes can stream at the same time to the output nodes which can give very fast retrieval times. This parallel processing cannot be performed by a conventional computer, which can only do one computation at a time (serial processing).



X_1	X_2	Y
0	0	0
1	0	1
0	1	1
1	1	1

Figure 5. A simple neural network simulating the logical OR-function. This function requires a '1' at the output if at least one of the input values is a '1' (see the table on the right). The output values are transformed to a binary code with the hard limiter shown on the left.

The efficient active memory structure that was demonstrated here, resembles the way human memory is organized. So it appears that the practical problems that arise by the use of word models in the Markov approach (lack of training data and (computer) memory capacity) do not apply to humans.
If we assume that words are recognized by humans without any intermediate units, this poses the following problem: How do we recognize the smaller units when these are spoken in isolation? An answer to this question might be that the acoustic input is simultaneously processed through separate channels, such as phoneme channels, syllable channels etc. These parallel processes work independently. The results of all parallel processing steps could be compared at a superior level. This might be done by chopping the unit at a higher level into smaller units of the lower level. For instance the badly pronounced word /bʌʃ/ might be recognized by the word processing channel as the word "bus" which can be chopped into the (ideal) phonemes /b/, /ʌ/ and /s/, whereas the phoneme processing channel would produce the phoneme string /b/, /ʌ/ and /ʃ/. In this way a listener could find out that the word "bus" was badly pronounced. The proposed theory of parallel channels can easily account for the processing of new words that have to be added to the lexicon. These unknown words are processed by the lower levels such as for instance the phoneme channel. The string of phonemes is subsequently passed on to the long term memory where it serves as an initial word template that can be further trained with other specimens of the word.

3.2 The variability in speech

There are many sources of variability in speech, such as speaking rate, speaking style, vocal tract length, dialect, etc. How do listeners cope with all this variability in speech? There are three options:

1. They construct a new template for every variant of a speech unit. This would lead to an almost infinite amount of templates.
2. They use one 'average' template for each speech unit and apply rules to account for all the variants of the unit. The main problem with this approach is to know which rule to use, if you don't know what has to be recognized. In other words, you have to recognize *before* you can apply the rule. Consider for instance the sentence "The typhoon came to New York". The /n/ at the end of "typhoon" may be changed to a /ŋ/ under the influence of the /k/ at the beginning of "came". This phenomenon can

be stated in an assimilation rule, but we should be well aware of the fact that such a rule is used to explain a sound shift *after* we have found out which words were involved. In for instance the sentence "The king came to New York" no assimilation has occurred. We are aware of this, because we know that the word "king" ends with a /ŋ/. For a recognition system such rules would only be useful if they could be applied in advance and not after the recognition was already done. As stated before, such rules arise from *reflection* on acoustic phenomena that we observe.

3. They use one 'average' template or only a small number of 'average' templates for each speech unit. This can be accomplished if the speech units are robust enough.

The third option appears to be the most attractive one, but can the units of speech be robust enough to cope with all kinds of variability? It will be clear that words are more robust units than for instance phonemes. A consonant that has been affected by assimilation will be recognized different from the consonant that was intended by the talker; a reduced vowel in a word will be easily confused with other vowels. However, a word like "typhoon" can still be easily recognized if the first vowel is reduced and the /n/ is replaced by a /ŋ/. The power in the strategy of hidden Markov modelling is that training emphasizes the more reliable parts of the word (for instance the stressed syllable) by means of strongly peaked distributions of acoustic events. These reliable parts are especially helpful to discriminate between words. The robustness of words in speech recognition can be compared with the robustness of large visual patterns. A face for instance can easily be identified under a lot of circumstances (laughing, crying, eating etc.). It can even be identified if only part of it is shown or if it is 'distorted' by a beard or glasses. In the same manner, a word can be recognized easily under a lot of circumstances, even if parts of it are missing.

Hidden Markov models are flexible enough to cope with different speech rates. Durations are namely not modelled by mean values and standard deviations, but by *probabilities* (see chapter 2). For a word model this means for instance, that as long as the durations of the different parts within the word remain unchanged *relative* to each other, recognition performance will not be affected. Hidden Markov models can also be trained for speaker-independent recognition. That is, single word models can be successfully trained for a large group of different speakers. This was shown by Lee (1989), who used hidden Markov models to train a lexicon of almost 1000 words. Training data consisted of a number of sentences in which these words occurred, read aloud by more than 100 different subjects. The recognition accuracy of the word models, tested with the sentences of 15 new subjects, was over 70%, based on acoustic evidence alone.

So it appears that probabilistic word models can be very robust with regard to different kinds of variability. They may even be robust enough to recognize speech properly without the need to normalize for different rates, speakers, etc., especially if we take the support from higher level knowledge into account (see section 3.4). To deal with very large differences between speakers, due to dialect, sex, etc., different sets of templates for separate speaker groups could be introduced. Support for the existence of such different sets of templates in the human mind was found by Van Bergem et al. (1988), who showed that humans probably use different templates to recognize the vowels from men and children.

Let us now return to the issue of nonsense words already mentioned in chapter 2. Given the enormous amount of variability in speech utterances, it is impossible to make a clear distinction between 'real' words and 'nonsense' words. Human listeners are very well capable to recognize a mispronounced word or a word that was partly masked by some kind of noise, although in a strict sense these words would have to be marked as 'nonsense'. In a normal communicative situation a listener will always try to find a match for the words he hears, even if they are unknown to him. This is done in the

Markov strategy by comparing the output probabilities of all possible words and picking out the most likely one. However, the 'distance' between the output probability of the recognized word and the average output probability of the word (see figure 3) gives the opportunity to establish the plausibility of the recognized word. If the plausibility is too low, a listener may wonder if he has heard the word correctly and ask the talker for a verification.

3.3 The processing of continuous speech

In the spoken sentence "I recognize speech" there are no silences between the words that could serve to detect begin points and end points. A segmentation of the sentence into words or smaller units prior to recognition is extremely difficult. Besides, a closer look at the string of sounds that composes this sentence reveals that several other words are embedded in it, for instance "wreck", "nice", "ice", "peach", or "each". How do listeners choose the right set of words from the abundance of possible words that are usually conveyed in a sentence?
In speech recognition with hidden Markov models, this problem is dealt with in a manner that is rather straightforward: The probability of all possible paths of all possible word sequences is calculated and the path with the highest probability is chosen. As shown in figure 6 on the left, the recognition search tries to determine the best path through a network of n words. Examples of such paths are shown on the right of figure 6.

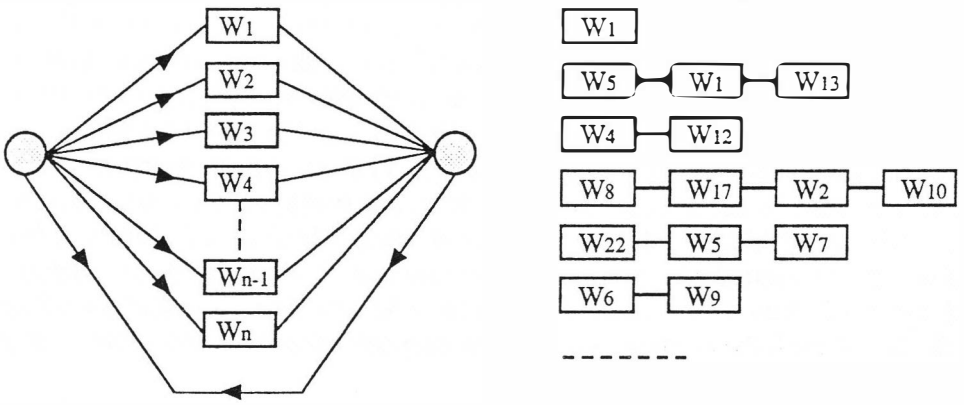


Figure 6. Recognition of continuous speech from a lexicon of n words. On the left a network with a feedback loop is shown that contains n words. Examples of possible paths through the network are shown on the right.

The lower bound to the number of word models in a string is one. The upper bound is determined by the total duration of the sentence and the minimal duration for each word model. In each candidate string the word models are matched to the acoustic data in the best possible (statistical) way. If a sentence contains for instance several words (in reality) and it is matched with only one word model, all acoustic evidence in the sentence would have to be divided over just this one model. Therefore, the output probabilities of the same word models in different strings may be very different.
This may seem a rather exhaustive way of recognizing a string of words, because the number of possible paths grows exponentially with the number of words in the lexicon. However, algorithms have been developed that give a drastic reduction in the number of necessary computations (e.g. a Viterbi search, see Lee, 1989) by only considering paths with the highest probability at each time step. These algorithms are very efficient

and robust and there is no need for segmentation beforehand. Furthermore, this strategy gives the opportunity to integrate higher level knowledge with the acoustic evidence. This point will be discussed in the next section.
The problem of words that are 'hidden' in other words, as mentioned at the beginning of this section, can be solved with the proposed strategy. For the sentence "I recognize speech" several strings are hypothesized that contain the word "speech" and others that contain the word "each". The acoustic evidence for both words is probably well matched with their corresponding models in these strings. However, the sentence part "I recognize sp" has to be accounted for as well in the strings containing the word "each". This part will probably poorly match any possible combination of words, so that the *overall* probability of all the strings containing "each" will be relatively low. Notice that this argument would not apply to the sentence "I recognize peach" . In this case higher level knowledge is needed to solve the acoustic ambiguity. In the next section we will discuss how this can be done.

3.4 The use of higher level knowledge in speech recognition

There are at least three possible ways of using higher level knowledge in the processing of sentences:

a. *After* the acoustic processing (1).
   The acoustic analysis (e.g. performed in several steps with intermediate speech units) could result in a first hypothesis of a word string that could be input to a higher level analysis. This might result in a rejection of the word string proposed by the acoustic analysis. In this case the acoustic analysis would have to be repeated, which would lead to a new hypothesis of a word string, which might again be rejected etc. (see figure 7a). The problem with this approach is how to perform the acoustic analysis after the first one, which was obviously done with the highest possible accuracy. Should we repeat the acoustic analysis with less accuracy? Which parts of the acoustic input should be labelled differently? Of course the acoustic data would have to be available for a long time if the acoustic hypotheses were rejected over and over again. In addition, it is not at all clear what criteria should be used by the higher levels to reject the acoustic analysis. Should we reject the proposed word string because the words have a low frequency of occurrence or because the words are unlikely in the given context? Should we reject the entire word string or only a part of it and if so which part?
b. *After* the acoustic processing (2).
   The string of acoustic events could be processed in the way described in section 3.3. The most probable string of words resulting from the acoustic analysis could be input to the higher level analysis. This might result in a rejection of the word string proposed by the acoustic analysis. In this case the second best word string found by the acoustic analysis could be input to the higher level analysis and so on, until the best compromise between all available knowledge sources would be reached (see figure 7b). In this approach all possible word strings that were analyzed at the acoustic level (or a subset) would have to be stored. Moreover, the higher level analysis would have to be done over and over again (just as in the previous proposal) and solid criteria for rejecting the acoustic analysis would have to be established.
c. *Simultaneously* with the acoustic processing.
   This could be accomplished by combining the acoustic probabilities that were obtained in the way described in section 3.3 with higher level probabilities (see figure 7c) in one analysis step.

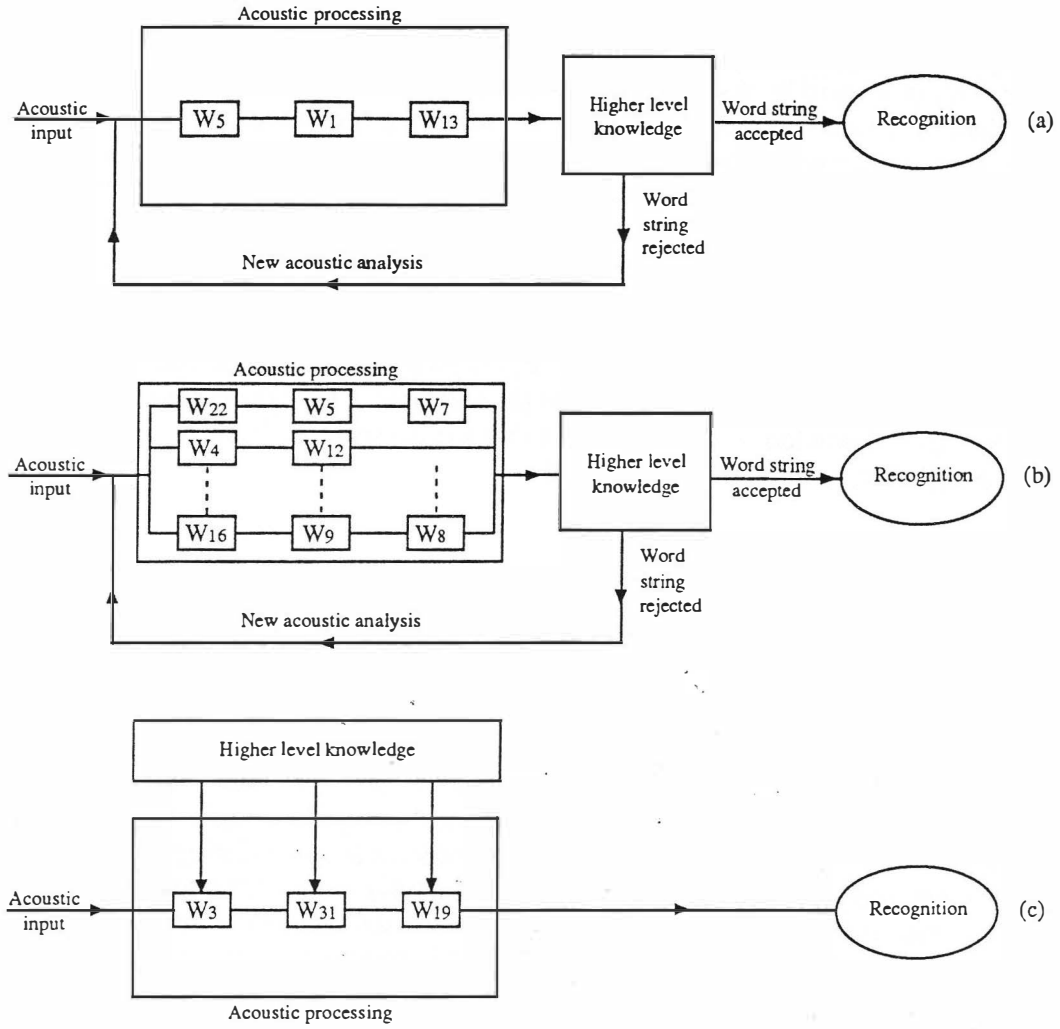


Figure 7. Three ways of recognizing a string of words. In (a) both the acoustic analysis and the higher level analysis are done over and over again. In (b) all acoustic alternatives are stored (sorted by likelihood) and subsequently tested against the higher level evidence one at a time. In (c) the most probable word string is produced based on the acoustic analysis and higher level knowledge simultaneously; there is no feedback.

The most efficient method of using higher level knowledge appears to be the last one. But how can we translate higher level knowledge into probabilities? To illustrate the way this can be accomplished, we consider four types of probabilities. Each type works on a different stretch of time:

1. Probability caused by frequency of occurrence. This probability is a permanent part of a word.
2. Probability caused by word order effects. The probability that for instance the article "the" is followed by a word other than a noun or an adjective is very small. The probability that a word is followed by itself is even smaller (at least in Western languages). Such a network of inter-word probabilities constitutes a permanent part of the lexicon.
3. Probability caused by the topic of conversation. A lecture about sailing boats may for instance raise the probability within several 'semantic fields' (containing words like "sea", "captain" etc.). This probability will last for the duration of the conversation (a few minutes to a few hours).

4. Probability caused by a triggering of words within the same 'semantic field'. If for instance the word "sit" is hypothesized in a sentence, it is likely that words like "chair" or "couch" follow. This probability will last only a couple of seconds.

The total probability of a candidate word string for a sentence can simply be calculated by multiplying the total acoustic probability with the total higher level probability. This results in just one word string with the highest *combined* probability (and a hierarchy of less probable word strings). Up till now only the higher level probabilities mentioned under (1) or (2) are often successfully used in hidden Markov models. Those mentioned under (3) and (4) require the construction of 'semantic fields' by connecting word concepts that are semantically similar in some way. Such a network could be build up from the moment that we learn our first words. The acoustic ambiguity, mentioned in the former section, can be solved with the probabilities mentioned under (3). If the topic of conversation would e.g. be 'food', then the probability of "peach" would be raised. If, on the other hand, the topic of conversation would be 'phonetics', then the probability of "speech" would be raised.
Postprocessing may occur at the level of interpretation (*reflection* on the recognized word string). At this stage of processing all earlier analysis steps may very well have been disposed of already, except for the final results. If for some reason the sentence "The hat climbed the tree" was heard, it may be substituted by "The cat climbed the tree", because of the acoustic similarity of "hat" and "cat" and the fact that it is implausible that a hat climbs a tree. Only problems in interpretation may lead to such a substitution of words. Similar adjustments can be made at this stage to grammatical errors in the message. In this view the only syntactic constraints that are used before the postprocessing are word order probabilities (the second component mentioned above).

4. CONCLUSION

In this article we have tried to show the efficiency and the plausibility of word recognition with probabilistic models. This was done by referring to the Markov approach in speech recognition. We think it very likely that human word recognition is also primarily of a probabilistic nature. If one asks somebody to estimate the probability of sunny weather in Holland or Italy, he will presumably be able to give you a reasonable guess. People also very well know that the chance to be hit by lightning is very small. The concept of probability is surely not unfamiliar to us. It is a very natural way of expressing the proportion of times a certain event *does* occur and *does not* occur. People can use this probabilistic knowledge about events to formulate explicit 'rules'.
We demonstrated how probabilistic strategies can be used to account for the variability in speech utterances, to isolate words from the continuous flow of speech, and to combine in an elegant way higher level knowledge with the acoustic evidence. Furthermore, we proposed a separate processing of speech at the level of words and at the lower levels (e.g. phonemes).
In what way the probabilistic strategies could be realized in our brain is as yet uncertain. Perhaps the artificial neural networks with their more probable and less probable connections between nodes may give a good reflection of reality. However, it was not the aim of this article to investigate memory structures, but merely to demonstrate the power of the probabilistic approach in word recognition.

ACKNOWLEDGEMENTS

I would like to thank Uli Frauenfelder, Louis Pols, Florien Koopmans-van Beinum, Gitta Laan and Louis ten Bosch for their useful comments on earlier versions of this article.

REFERENCES

Alphen, P. van & Bergem, D.R. van (1989). "Markov models and their application in speech recognition", Proc. of the Institute of Phonetic Sciences Amsterdam 13, 1-26.

Bergem, D.R. van, Pols, L.C.W. & Koopmans-van Beinum, F.J. (1988). "Perceptual normalization of the vowels of a man and a child in various contexts", Speech Communication 7, 1-20.

Bergem, D.R. van (1990). "The influence of linguistic factors on vowel reduction", Proc. Linguistics and Phonetics, Prague (in press).

Cole, R., Phillips, M., Brennan, B. & Chigier, B. (1986). "The CMU phonetic classification system", Proc. ICASSP '86, Vol.3, 2255-2258.

Elman, J.L. & Zipser, D. (1988). "Learning the hidden structure of speech", J. Acoust. Soc. Am. 83, 1615-1626.

Elman, J.L. (1989). "Connectionist approaches to acoustic/phonetic processing" In: Marslen-Wilson, W.D. (Ed.), Lexical representation and process, 169-226.

Fant, G. (1990). "Speech research in Perspective", Speech Communication 9, 171-176.

Forster, K.I. (1979). "Levels of processing and the structure of the language processor", In: Cooper, W.E. & Walker, E.C.T. (Eds.), Sentence processing: Psycholinguistic studies presented to Merrill Garrett, Hillsdale, N.J.: Lawrence Erlbaum Associates, 27-86.

Jelinek, F. (1985). "The development of an experimental discrete dictation recognizer", Proc. IEEE 73, 1616-1624.

Klatt, D.H. (1989). "Review of Selected Models of Speech Perception", In: Marslen-Wilson, W.D. (Ed.), Lexical representation and process, Cambridge, Mass.: MIT Press, 169-226.

Lee, K.F. (1989). "Automatic Speech recognition: The Development of the SPHINX System", Kluwer Academic publishers, Boston.

Marslen-Wilson, W.D. & Welsh, A. (1978). "Processing interactions and lexical access during word recognition in continuous speech", Cognitive Psychology 10, 29-63.

Marslen-Wilson, W.D. (1987). "Functional parallelism in spoken word recognition", Cognition 25, 71-102.

McClelland, J.L. & Elman, J.L. (1986). "The TRACE model of speech perception", Cognitive Psychology 18, 1-86.

Morton, J. (1982). "Disintegrating the Lexicon: An Information Processing Approach", In: Mehler, J., Walker, E.C.T. & Garrett, M. (Eds.), Perspectives on mental representation, Experimental and Theoretical Studies of Cognitive Processes and Capacities, Hillsdale, N.J.: Lawrence Erlbaum Associates, 89-109.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986). "Learning internal representations by error propagation", In: Rumelhart, D.E. & McClelland, J.L. (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Cambridge, MA: MIT Press, 318-362.

Zue, V.W. (1985). "The use of speech knowledge in automatic speech recognition", Proc. IEEE 73, 1602-1615.