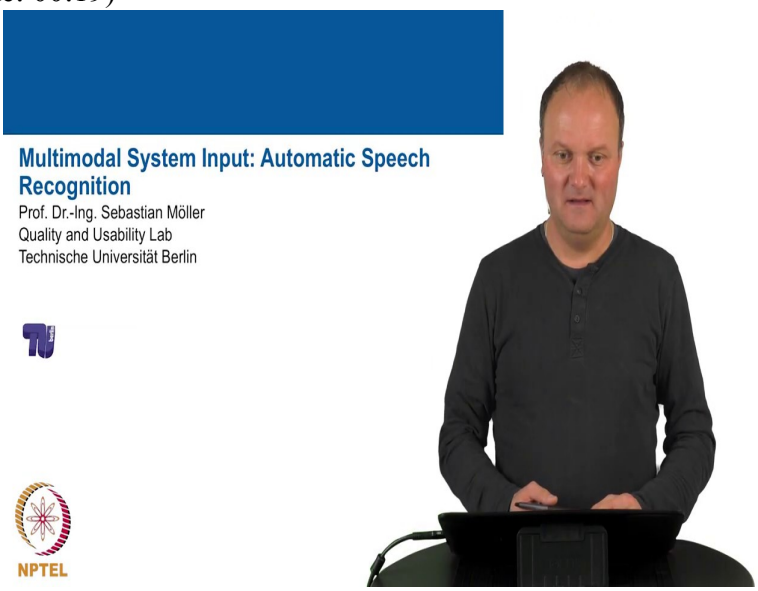
Multimodal System Input
Professor Doctor Sebastian Möller
Quality and Usability Lab
Technische Universitäät Berlin
Automatic Speech Recognition

(Refer Slide Time: 00:19)

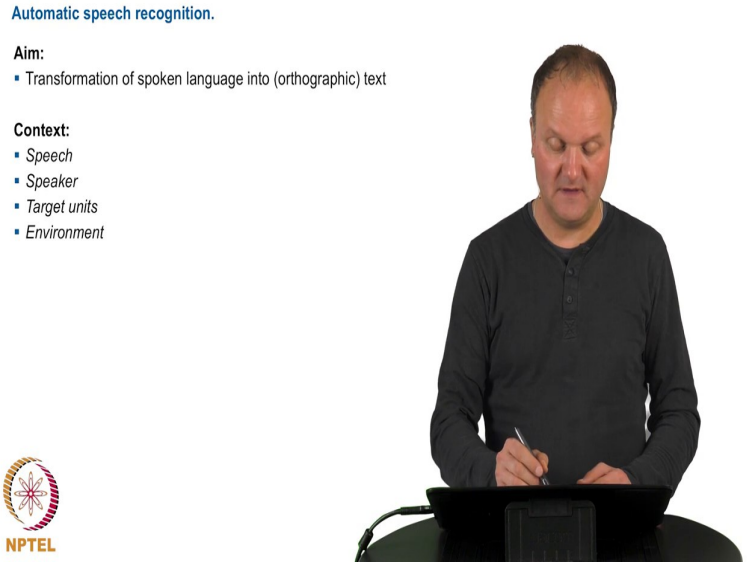


Following videos will talk about automatic speech recognition. Actually automatic

(Refer Slide Time: 00:25)

speech recognition is the process which transforms spoken language into a text which is usually of autographic form. In order to do so, a couple of considerations

(Refer Slide Time: 00:36)



have to be taken into account which can be summarized under the ~~general (())~~term (00:40) context.

For example what type of speech needs to be recognized? It can be differentiated according to the language or would just use, perhaps a dialog, perhaps a certain speaking style. A human listener is able to distinguish these points. A machine has to be taught how to do that.

Then the information comes from different speakers. Speakers are either known to the system or not known to the system or the system may be able to adapt to a certain speaker by training or adapting its internal features.

You might even have to deal with uncooperative speakers in terms that you have a forensic speech recognition, that is the speaker is not necessarily aware that the system is trying to transcribe what he has spoken.

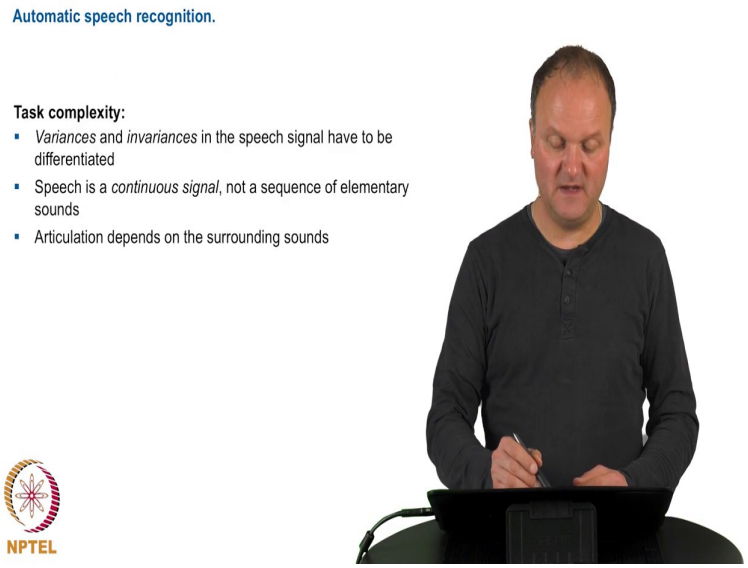
Speech recognition is differently difficult according to the target ~~audience~~ unit we are after. The number, the type of target units and the complexity plays a role. For example if you want to recognize just the digits between 0 and 9, this is an easier task than recognizing continuously spoken German or English speech.

And then of course the environment plays a role. For example if there is background noise, if there are transmission channels like telephone channels involved and so on, and all of ~~these~~

this taken together makes automatic speech recognition a very, very difficult task for a computer.

Actually there is a fundamental

(Refer Slide Time: 02:08)



difference between automatic speech recognition carried out by a computer and the speech recognition carried out by humans. We as humans are able to distinguish between the variances and the invariances in the speech and in the speech signal and this has to be explained to the computer, this has to be learnt by the computer.

So the learn, the computer needs to learn which are the invariant features which for example are representative ~~on ah, on /a/~~for an /a/ or an /e/ sound. We as humans have learnt that and can distinguish ~~ah~~ /a/ or /e/~~/a/~~ sounds coming from different speakers pronounced in different environments and in different languages and so on. And this has been taught to the computer.

The second difficulty is that speech is a continuous signal, that is we do not talk with individual words but we speak continuously and this continuous speaking style makes it difficult to separate the different units from each other.

Actually that is what we call the effect of co-articulation which means that one sound is linked to the other sound and the movements

(Refer Slide Time: 03:21)

of my articulatory organs span over from ~~the~~ one sound to the next and to the following one.

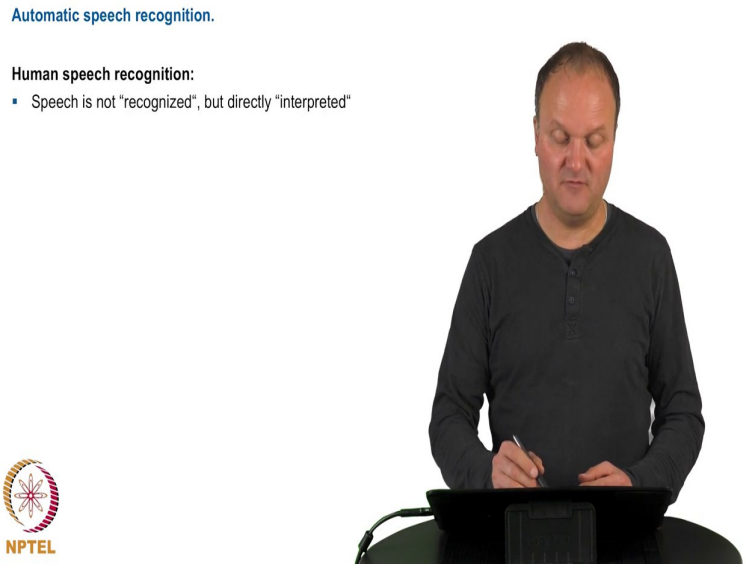
That is, if you want to recognize speech we initially do not recognize individual sounds but we recognize combinations of two sounds, diphones or three sounds, triphones. That is we try to model triphones in a computer in order to recognize that.

These effects which I call task complexity makes speech a very, very difficult task. And this can be seen, if you for example utter the three words to recognize speech or the five words to wreck ~~recognize~~ a nice beach. If I talk, if I speak them continuously you would n' ~~no~~t see much difference between to recognize speech and to ~~recognize~~ wreck a nice beach.

As an English-speaker you would of course know that, only the first, to recognize speech makes sense and to ~~recognize~~ wreck a nice beach does not make sense. But t~~T~~his is your interpretation of it and that' ~~i~~s what you as a human actually ~~doing~~do, you ~~are trying~~try to interpret speech ~~, viewing (()) (04:30)~~during listening.

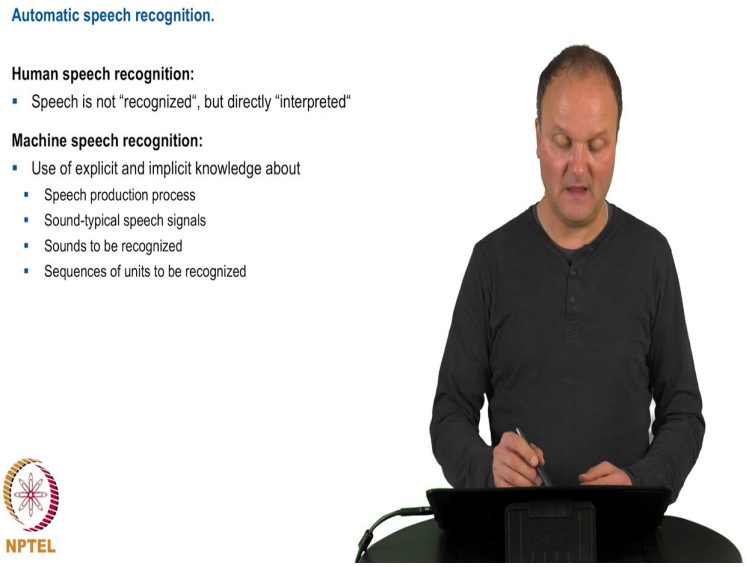
This is something which is not done by the computer. The c~~C~~omputer first transcribes speech and then tries to make out sense of what has been transcribed. So there are fundamental

(Refer Slide Time: 04:42)



differences in the processes.

(Refer Slide Time: 04:45)

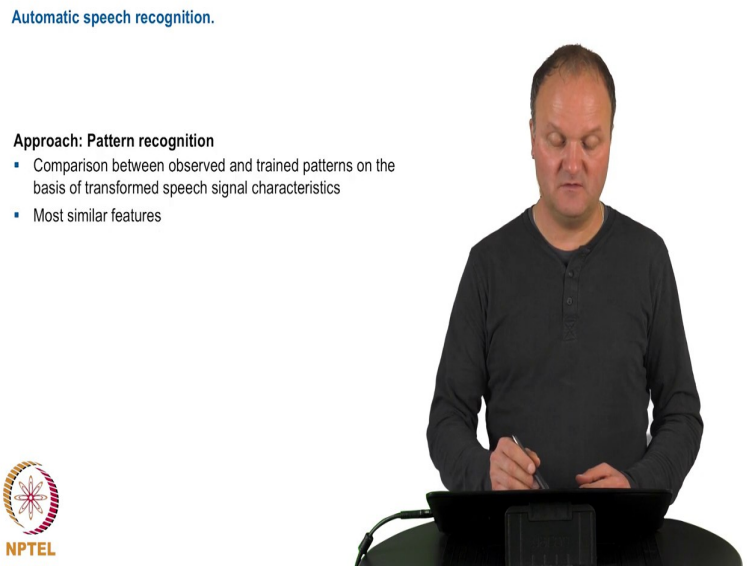


~~However~~However, the machine, the computer can make sense of background knowledge. The background knowledge could span over the speech production process that is how speech is produced by human and what the resulting characteristics in the speech signal are.

The machine could make use of typical examples of speech signals provided usually using training material. Then of course it needs to have an inventory of the sounds speech is composed of and it can recognize and then also an inventory of sequences of units to be recognized like a dictionary and a grammar which link these units.

This is our knowledge which we have to bring into the machine, to bring into the computer in order to automatically

(Refer Slide Time: 05:34)



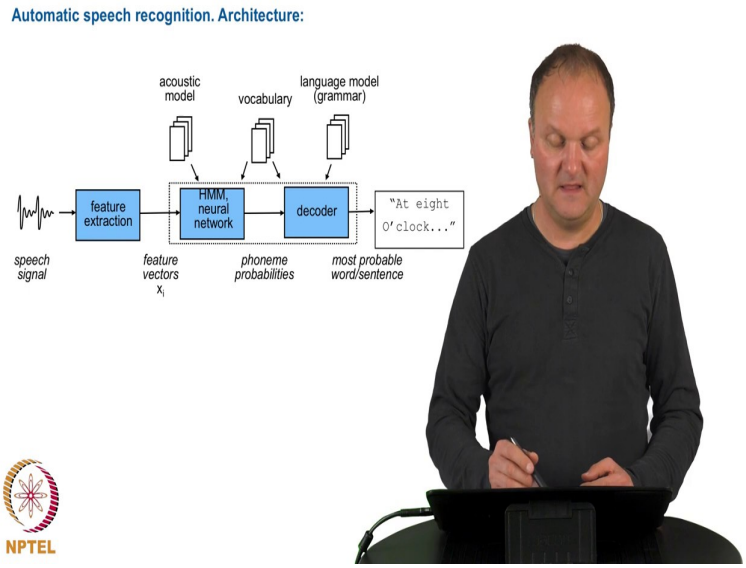
recognize speech.

The approach which is performed by the computer is a pattern recognition approach; that is the computer; the computer tries to compare trained patterns with what it can observe from the speech of the user which needs to be recognized.

And this is not done on the level of the physical speech signals that is the microphone signal. But this microphone signal is first transformed into a set of features which can easily be recognized.

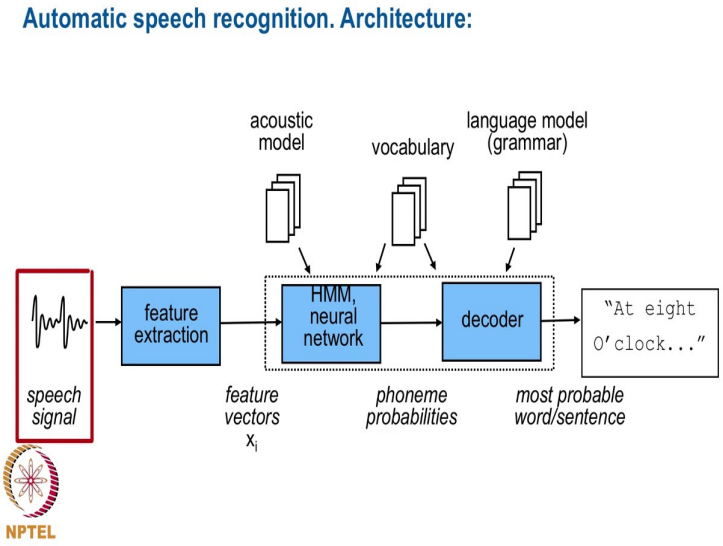
The recognition result would then be the most similar features to the ones which are already trained in the computer and the ones trained in the computer have a label with respect to what text has produced these features, and then this text is actually the recognition result. It is the most probably, the most probable hypothesis of the speech recognizer.

(Refer Slide Time: 06:38)



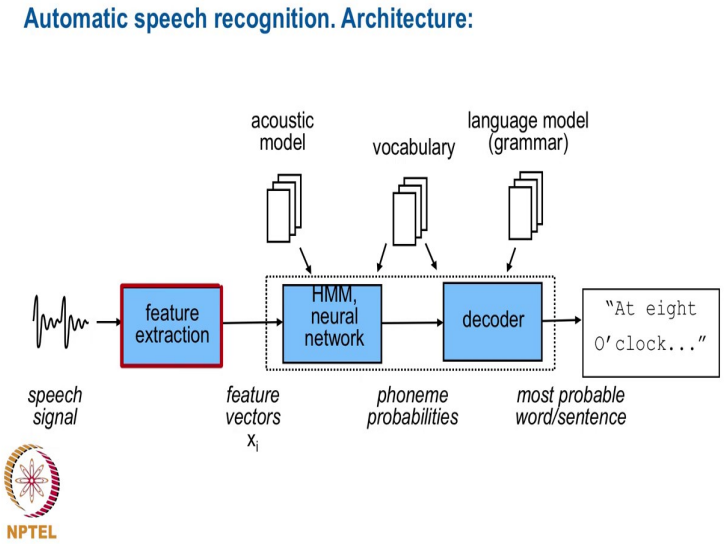
This pattern recognition approach can in a sum, be illustrated as you see in thea picture behind me. So the speech signal is

(Refer Slide Time: 06:46)



first transformed into features by means of

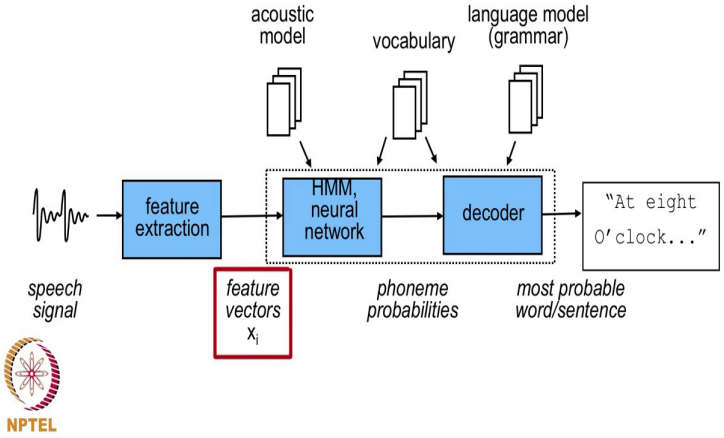
(Refer Slide Time: 06:49)



a features extraction algorithm. These features are usually calculated for time spans of approximately 20 to 30 milliseconds.

That is for each 20 or 30 milliseconds we calculate one vector

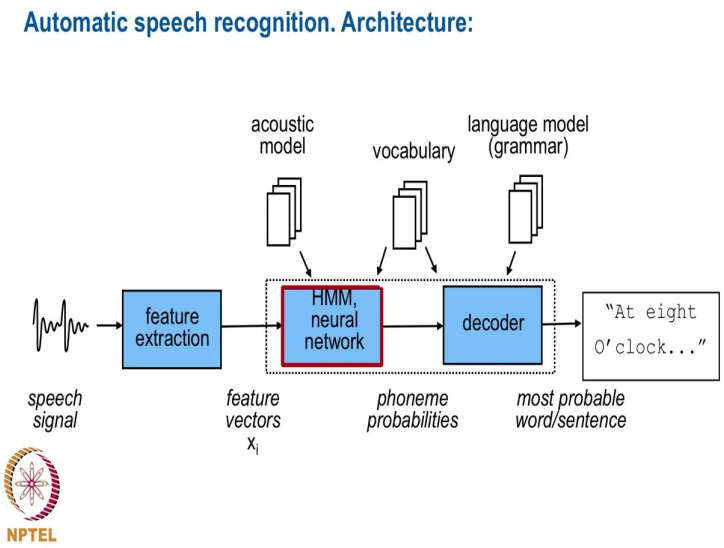
(Refer Slide Time: 07:02)



of features, usually about 20 to 30 features and these features then make reference to some sound or a phoneme, a class of sounds.

In order to associate the features with the phonemes in a probabilistic way, there are different mechanisms, different statistical algorithms. The most popular ones used nowadays in automatic speech recognizers are the so-called Hidden Markov Models

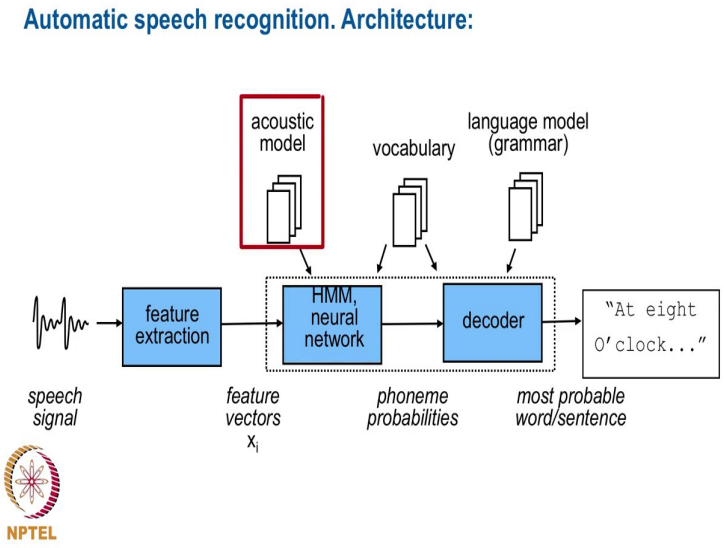
(Refer Slide Time: 07:32)



or H-M-Ms or the Neural Networks N-Ms.

In order to do so, the speech recognizer needs to be trained and needs to store which types of features are representative for certain phonemes and this is called the acoustic model

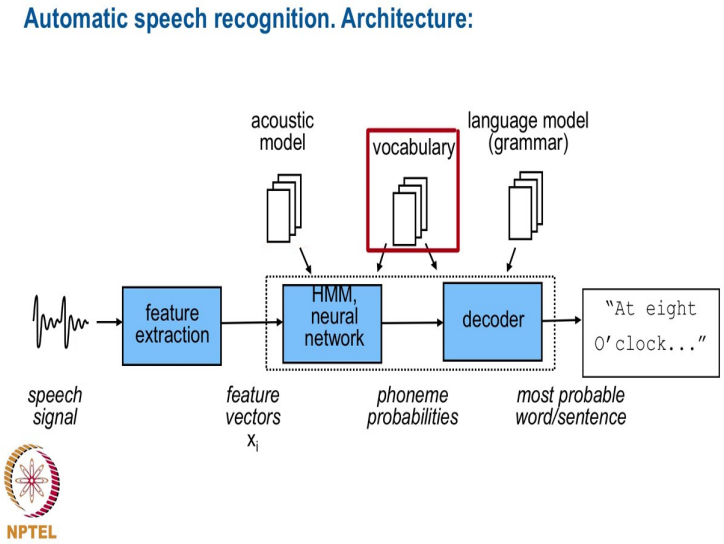
(Refer Slide Time: 07:49)



of thea speech recognizer.

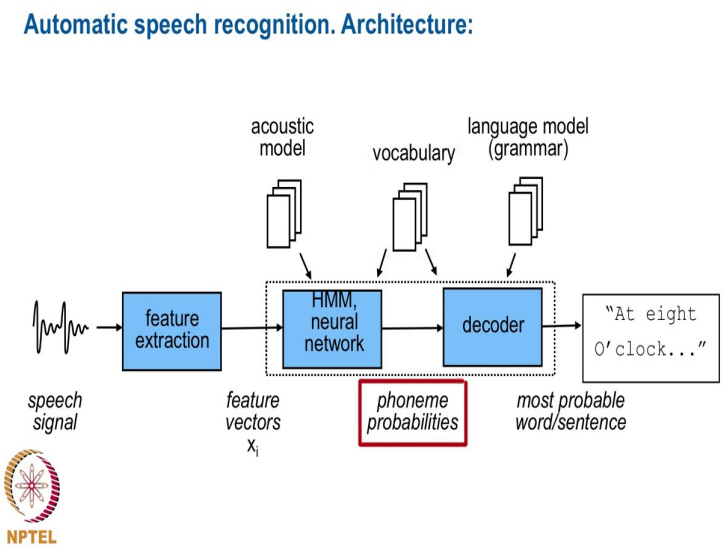
And the recognizer needs to know of course also which words can be

(Refer Slide Time: 07:56)



recognized. So it needs to have a vocabulary. The phoneme probabilities

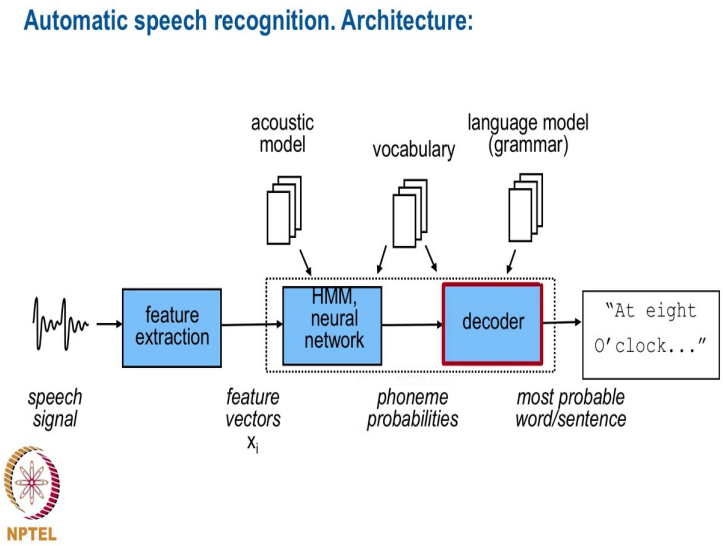
(Refer Slide Time: 08:02)



~~or~~ are the sequences of probable phonemes need then to be transformed into a sequence of words or even a sentence which is to be recognized.

In order to do so, there is a second step which is usually called the decoding step.

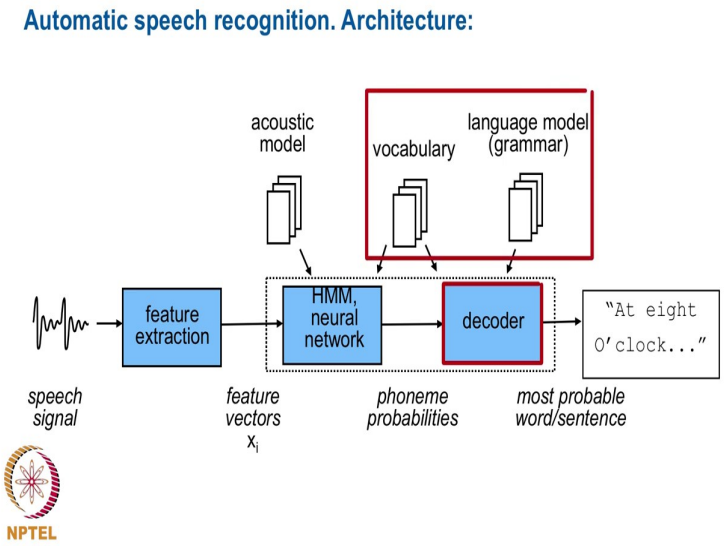
(Refer Slide Time: 08:17)



So in the decoder the phoneme probabilities are organized in order to form words and sentences.

In order to do so, the decoder needs to have access of course to the vocabulary but it also

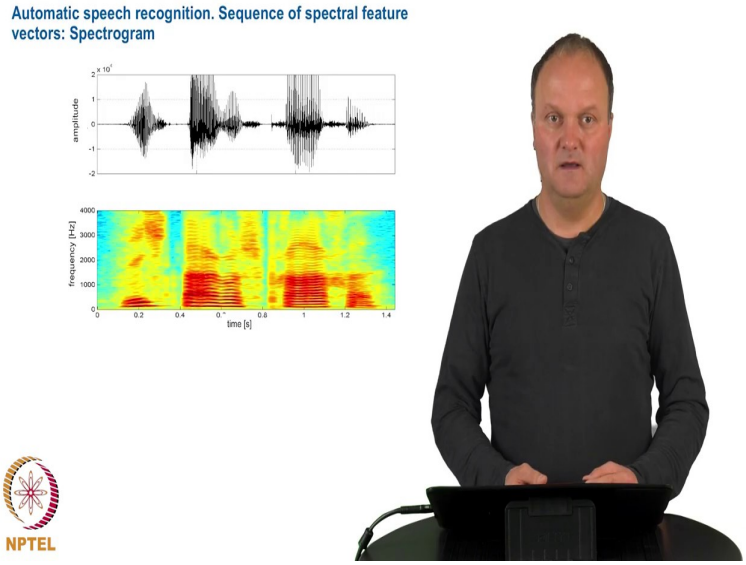
(Refer Slide Time: 08:28)



needs to know which words can follow each other, that it, that is it has to have a language model or a grammar in order to organize the words in a certain sequence.

The second step, the decoding step is usually also performed using a Hidden Markov Model, a different type of Hidden Markov Model or also a Nneural Nnetwork which ~~is becoming~~has become more popular in recent times.

(Refer Slide Time: 08:53)



A popular way to illustrate these features can be seen in the slide behind me. You see first, on the upper panel a picture of a speech signal that is the course ~~(()) (09:05)~~ of the microphone signal over time and in the lower part you see a spectral decomposition of thea~~a~~ speech signal into different frequencies from 0 to 4000 Hertz on the y axis and with the color coding you see the spectral energy or spectral power density at each time and each frequency point.

In principle this spectral representation gives you already a very nice representation in terms of features which could be recognized and in fact this so-called spectrogram has been used to make speech visible. So in principle we could apply an image recognizer on these types of features.

In reality

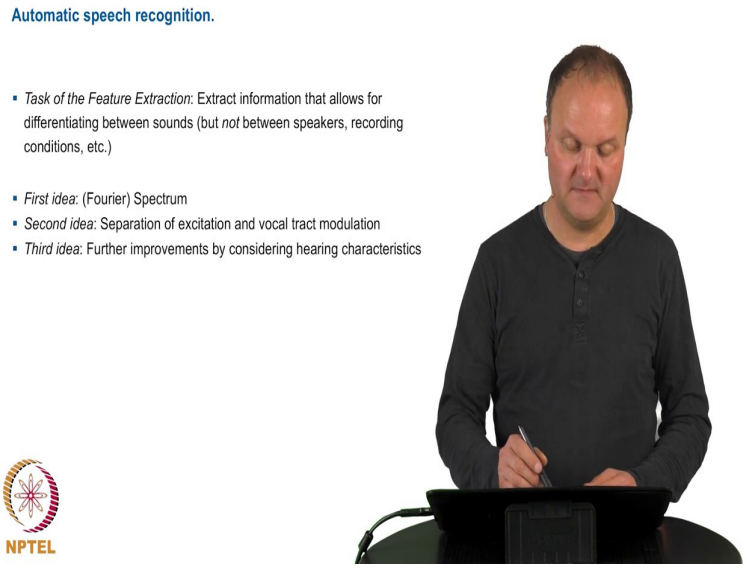
(Refer Slide Time: 09:47)

there are little bit more elaborated ways to calculate features from the speech signal namely to concentrate on the information which is really representative for the sound classes to be recognized that is for the phonemes but not to differentiate between the speakers, between recording conditions and so on.

So we need to only consider those types of information which are representative for the linguistic information in the speech signal and all other things have to be tuned off. So the Fourier spectrum is ~~the~~ a first approach state ~~(()) (10:24)~~ but it considers all types of information in the same way.

So we can see also background noise in the spectrogram. We can also see the fundamental frequency of the speaker, which might be an indication of whether it is a male or female speaker. All this needs to be thrown out of the feature representation.

(Refer Slide Time: 10:42)



In order to do so there are some other ideas which are usually combined in order to extract nice features which work well for automatic speech recognition processes.

First we try to separate the excitation signal and the vocal track modulation from the speech and then we usually make use of the hearing characteristics of the human ear in order to process speech in a way which is similar to what we have in the human ear.

And this may result in different types of representations; the most popular is the Mel-scaled Cepstral Coefficients or M-F-C-Cs, the second very popular type of feature is the Perceptual Linear Predictive or P–L–P coding and the third type of features is the Relative Spectral analysis or RASTA analysis.

All three result in features which represent short segments of the speech, for example 20 to 30 milliseconds long segments of speech.

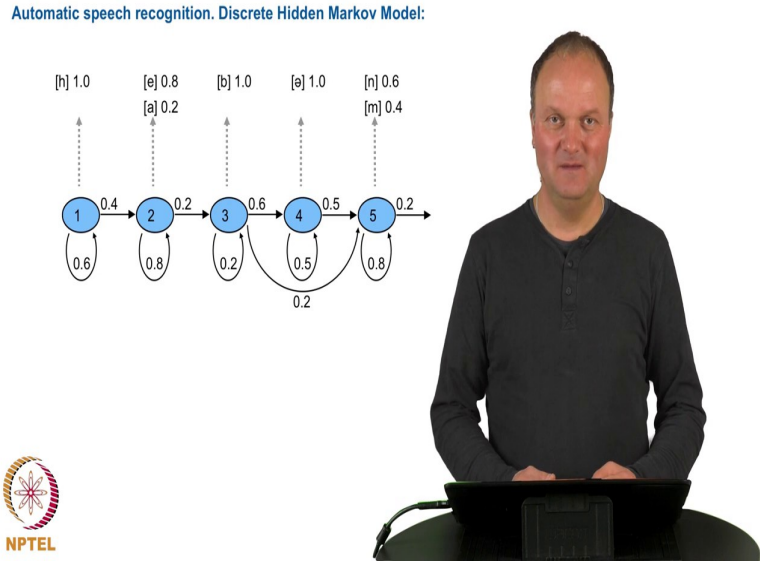
(Refer Slide Time: 11:50)

The next step is now to classify the features and if they are classified into phonemes, then to organize the phonemes and classify them into words or sequences of words.

There are two popular approaches which are used nowadays for this task. The first one is called the Hidden Markov Model and the next one is called the Nneural Nnetwork. Both of them are statistical approaches, that is they calculate a probability that a certain feature vector stems from or is related to a certain phoneme.

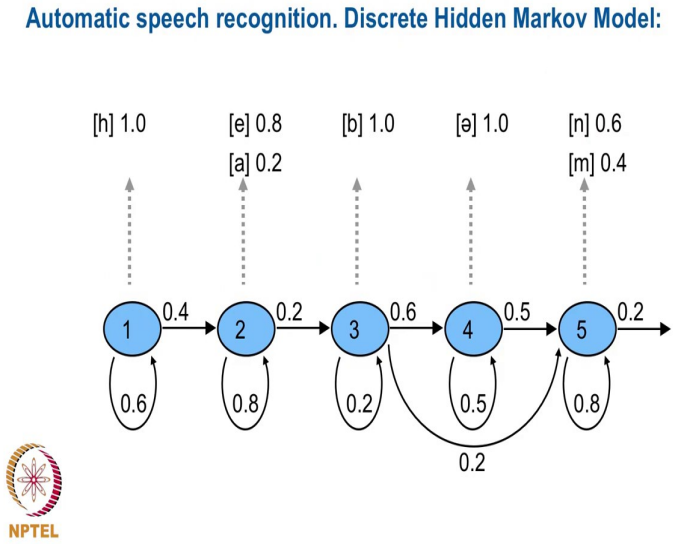
So it gives us probabilities of phonemes and these probabilities of phonemes can then be organized into probabilities of words. So all this is a probabilistic process.

(Refer Slide Time: 12:34)



Behind me you are seeing anthe illustration of the first type of Hidden Markov Model.

(Refer Slide Time: 12:39)



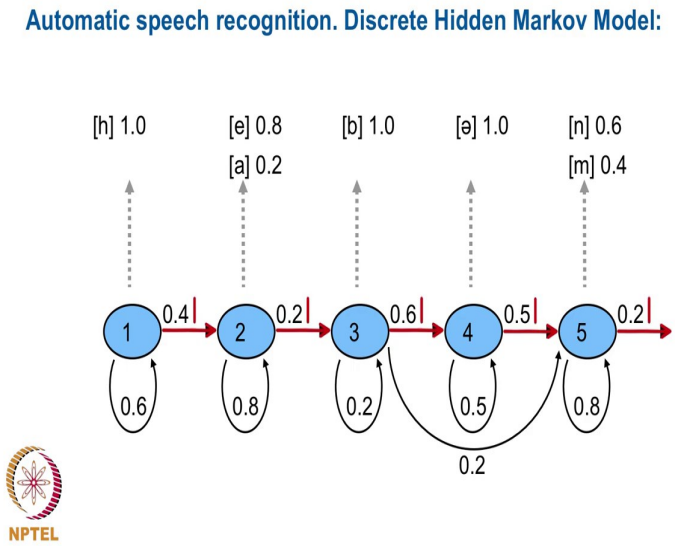
~~The next step is now to classify the features and if they are classified into phonemes, then to organize the phonemes and classify them into words or sequences of words.~~

~~There are two popular approaches which are used nowadays for this task. The first one is called the Hidden Markov Model and the next one is called the neural network. Both of them are statistical approaches, that is they calculate a probability that a certain feature vector stems from or is related to a certain phoneme.~~

~~So it gives us probabilities of phonemes and these probabilities of phonemes can then be organized into probabilities of words. So all this is a probabilistic process.~~

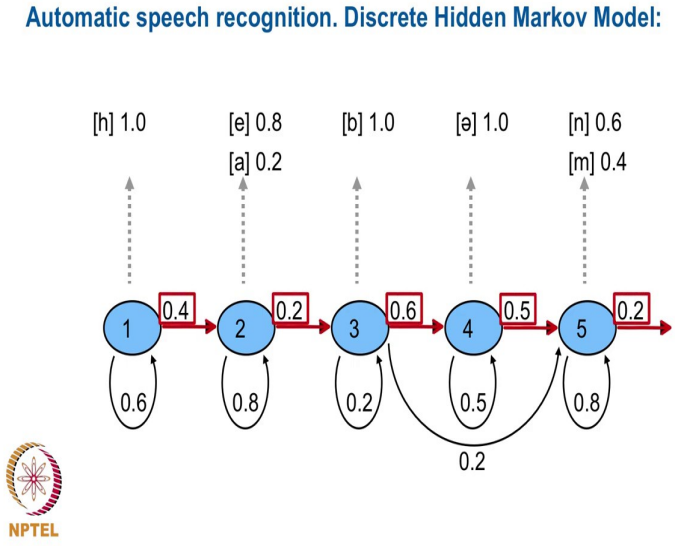
In principle, it consists of a series of states; these are these blue ovals which are connected by arrows, so-called transitions.

(Refer Slide Time: 12:51)



These transitions happen

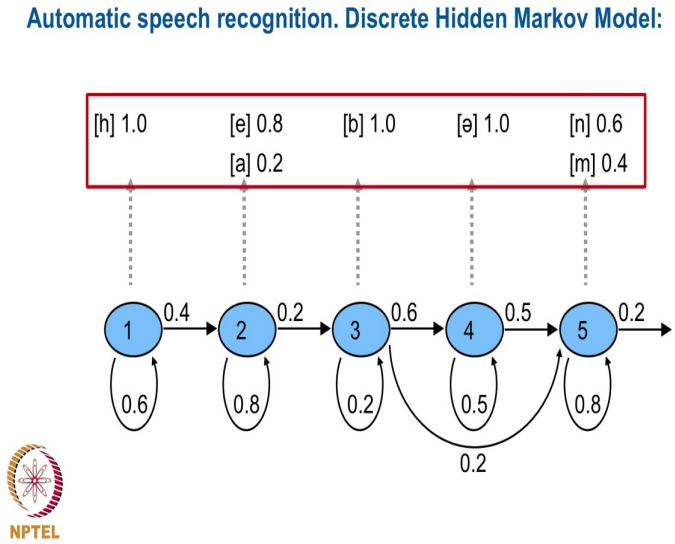
(Refer Slide Time: 12:52)



with a certain probability and this probability is indicated beneath the transition.

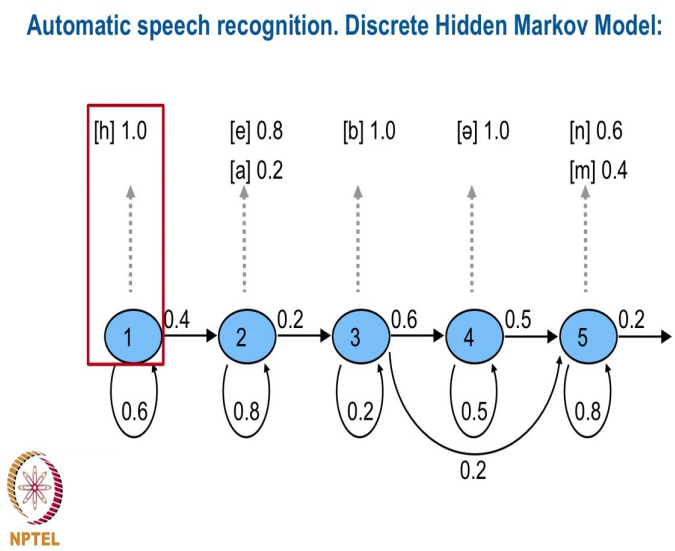
The states are also able to transmit or emit a certain information item,

(Refer Slide Time: 13:03)



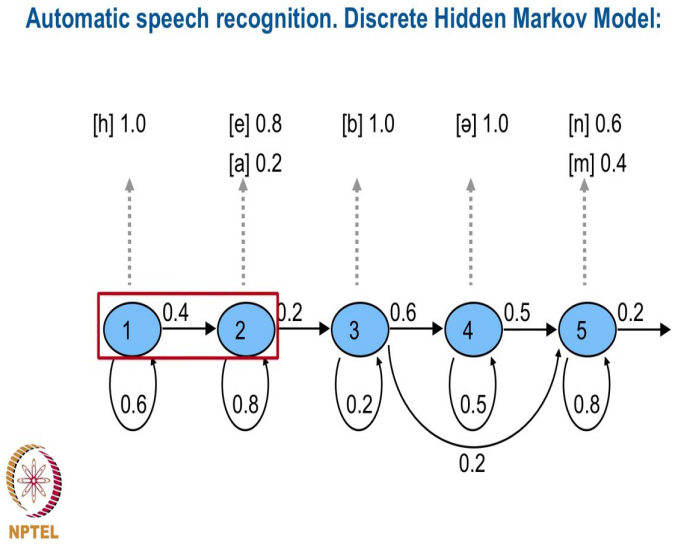
in this case a symbol and the probability with which it emits the symbol is also illustrated next to the symbols. For example, the probability that the state number 1

(Refer Slide Time: 13:18)



emits the symbol [h]~~h~~ is 1.~~1 point~~ 0. And the probability to transit from state 1 to state

(Refer Slide Time: 13:25)

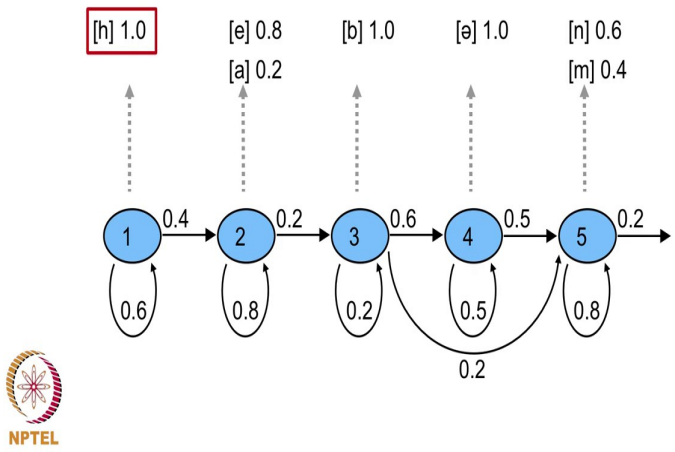


2 is actually 0 ~~point~~. ~~-~~4.

Now with the help of these probabilities one can calculate an overall probability that the sequence of ~~steps~~ states from 1 to 5 has emitted a certain sequence of symbols like~~. Like~~ the~~a~~ German word "heben" ~~heban~~.

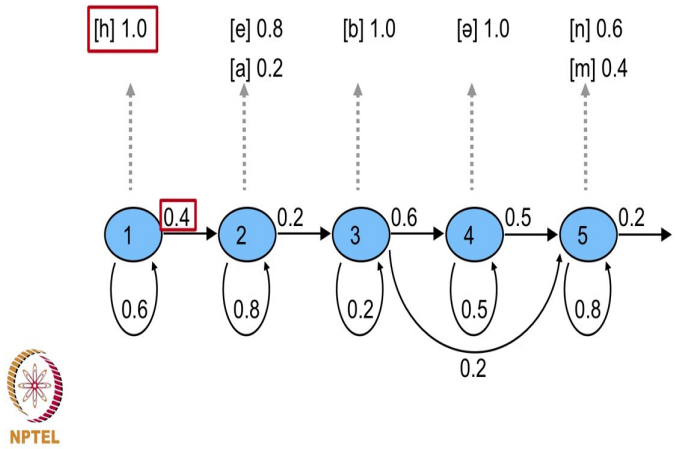
In order to calculate this probability we start with the state number 1 and calculate the probability that this state emits the symbol [h] which is 1~~. point~~ 0,

(Refer Slide Time: 13:54)



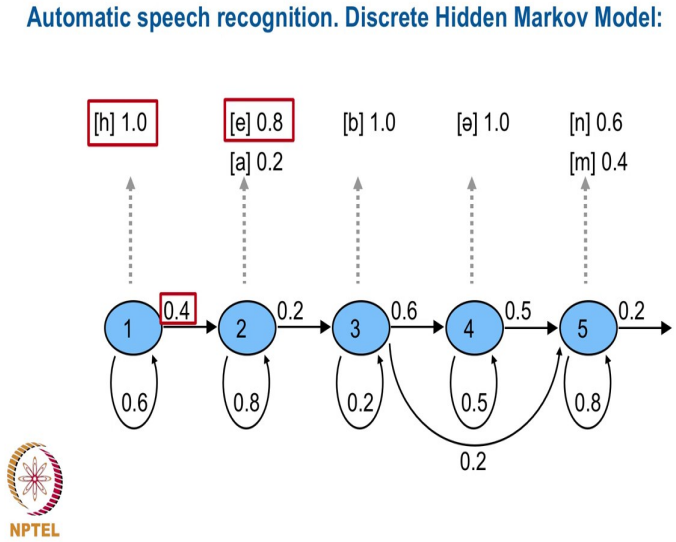
multiply it with the probability to transit from state 1 to 2, so

(Refer Slide Time: 13:59)



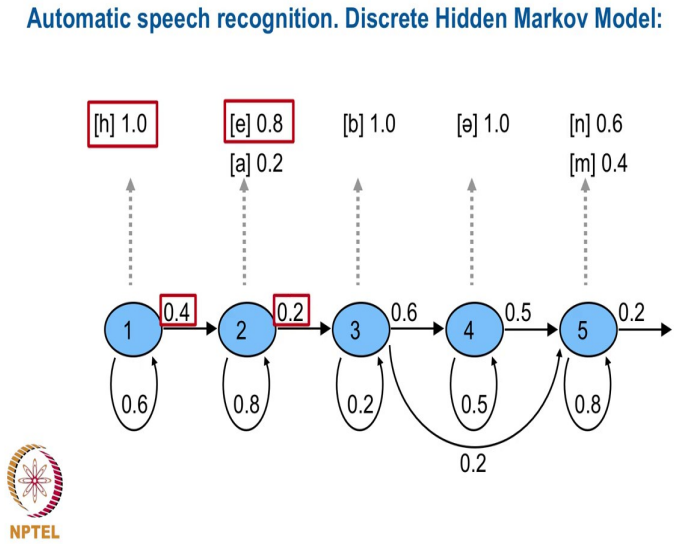
1 point. 0 multiply itied with 0 point. 4 multiply itlied with 0 point. 8

(Refer Slide Time: 14:05)



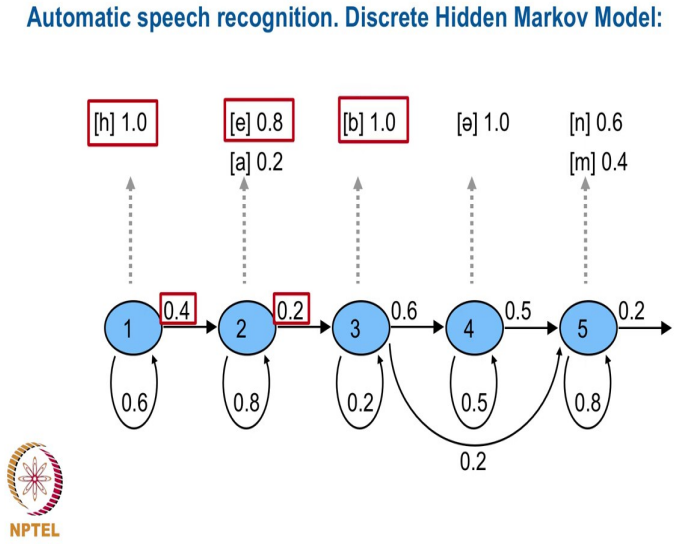
for emitting [e] multiply itied with 0 point. 2 to

(Refer Slide Time: 14:09)



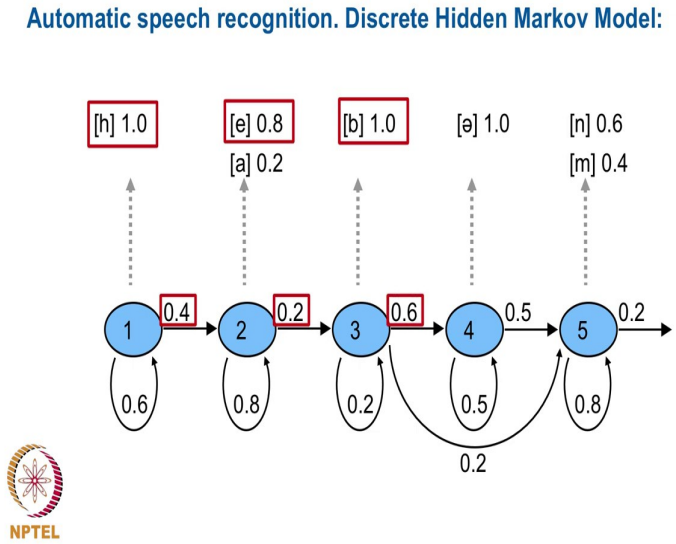
transit to state 3 multiply itied with 1 point. 0

(Refer Slide Time: 14:12)



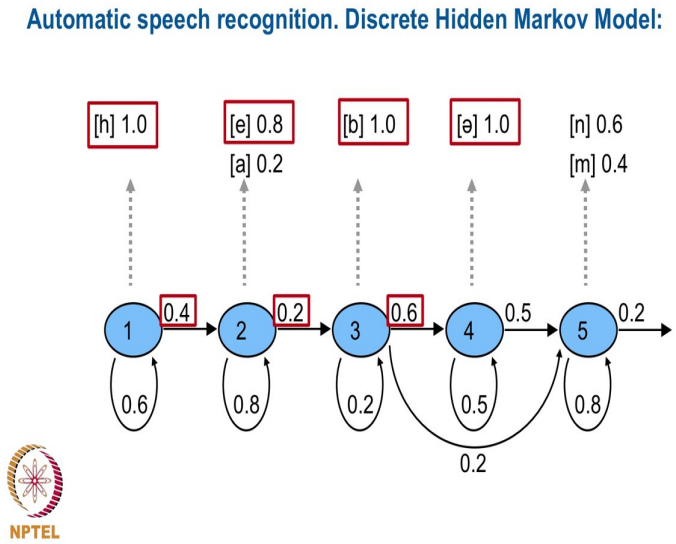
to make emit the [b]/b multiply itied with by 0. point 6

(Refer Slide Time: 14:16)



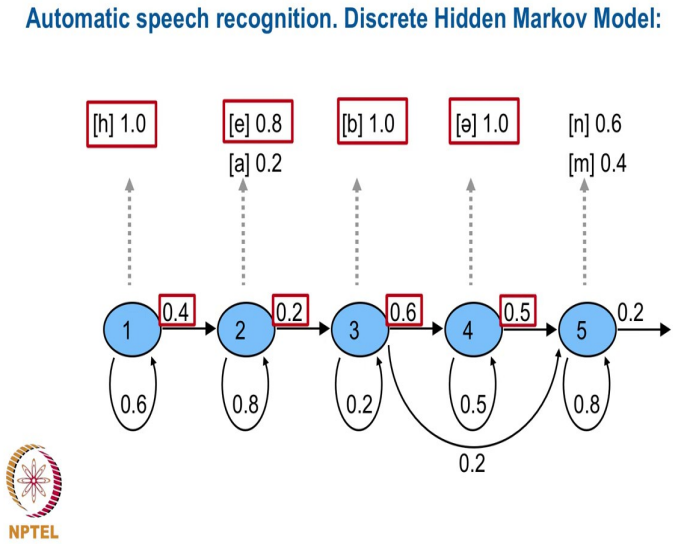
to transit to state 4 ~~multiplied~~ multiply it with 1~~. point~~ 0 for emitting

(Refer Slide Time: 14:20)



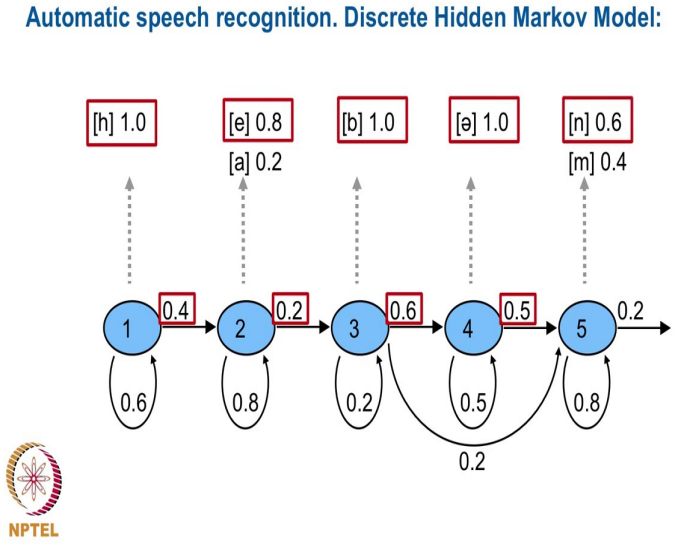
[ə] ~~the ah~~ multiply it~~ied~~ with 0~~. point~~ 5 to transit

(Refer Slide Time: 14:23)



to state 5 multiply itied with 0. point 6

(Refer Slide Time: 14:27)

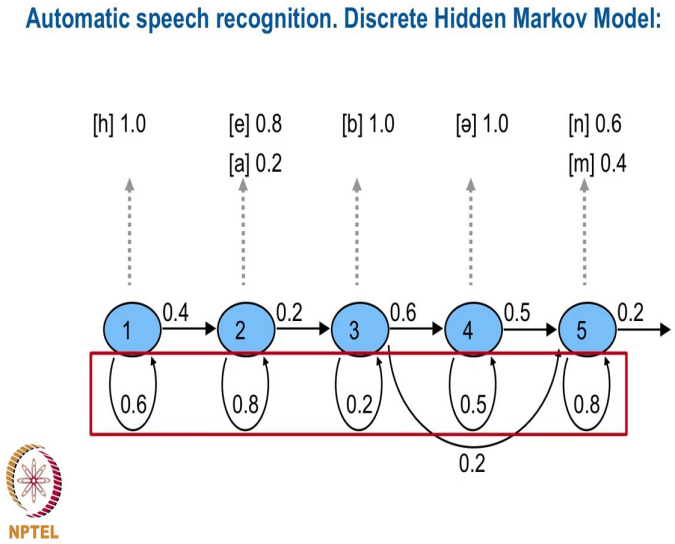


to emit the symbol [n].

So all this multiplication would result in the overall probability for this sequence of state emitting the symbol, the sequence of symbols "hebean".

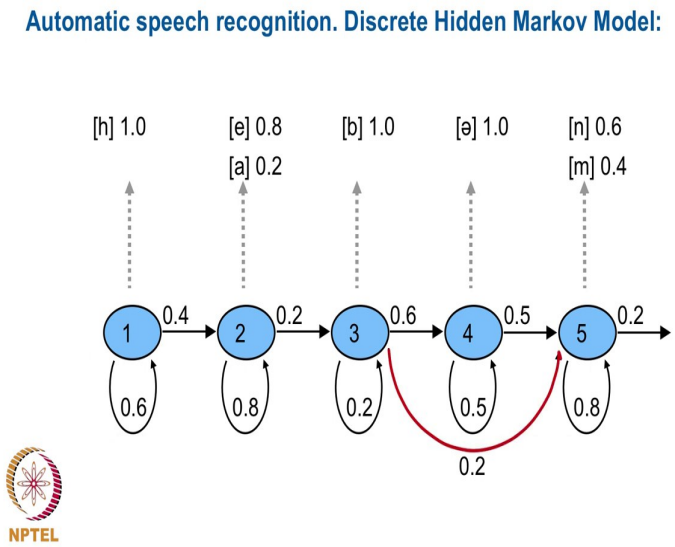
You are seeing that you can also stay in one

(Refer Slide Time: 14:44)



state for certain period of time then you transit from the state to itself and you can even bypass some

(Refer Slide Time: 14:51)

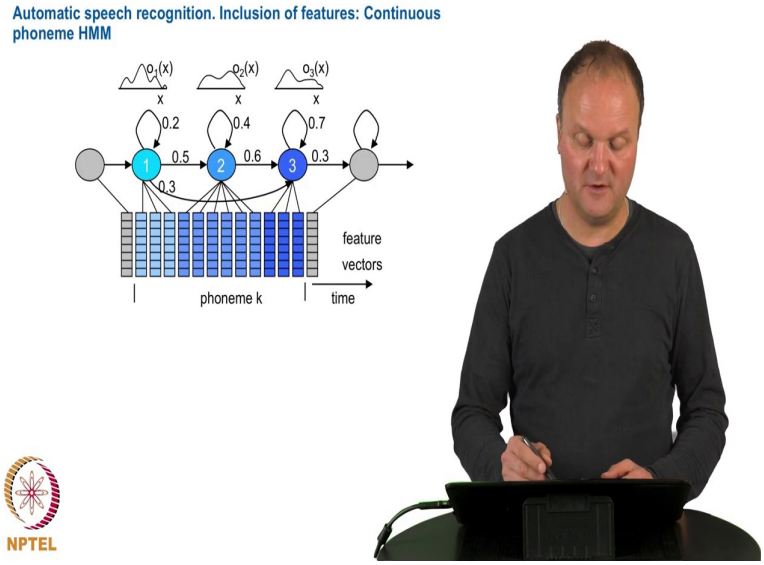


states as you see here. But Hidden Markov Model only allows you to go from left to right in emitting ~~signals~~symbols.

This type of Hidden Markov Model is useful for connecting information which comes ~~from~~ for example from the symbols in order to connect them to words or sentences but it does not consider the features which are necessary at the input of the recognition process.

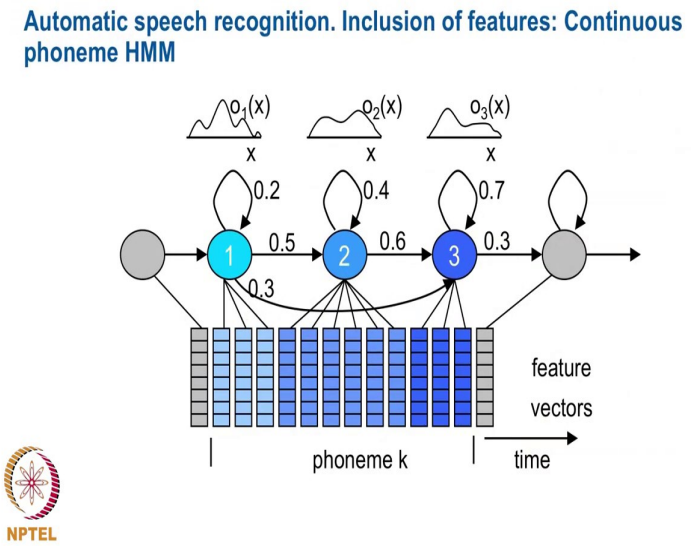
In order to deal with features, the second type of Hidden Markov Model,

(Refer Slide Time: 15:20)



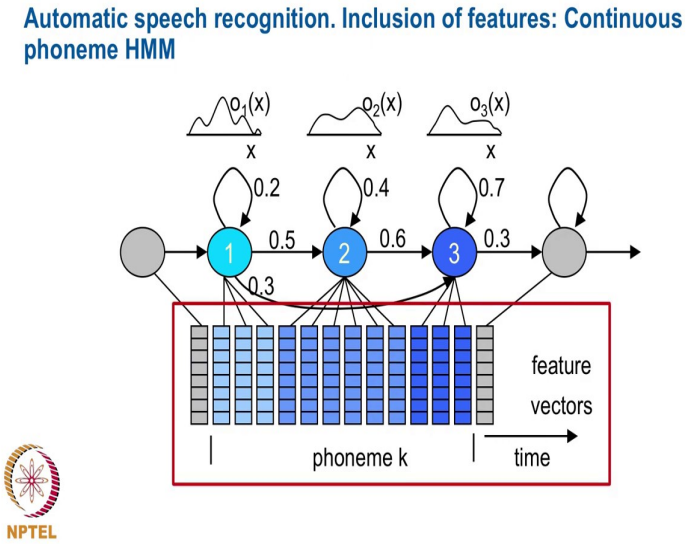
not a discrete one but a continuous phoneme Hidden Markov Model is used. An illustration of this Markov model can be seen behind me

(Refer Slide Time: 15:30)



and you see in the lower half of the picture,

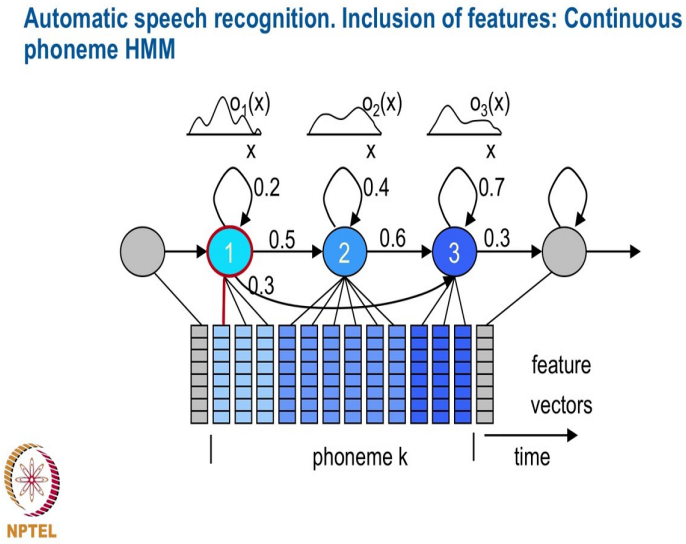
(Refer Slide Time: 15:32)



a sequence of feature vectors which are representative for a certain phoneme.

So there is a fixed relationship between the phoneme /k/ and the corresponding sequence of states. The link between the states and the feature vectors is a probabilistic

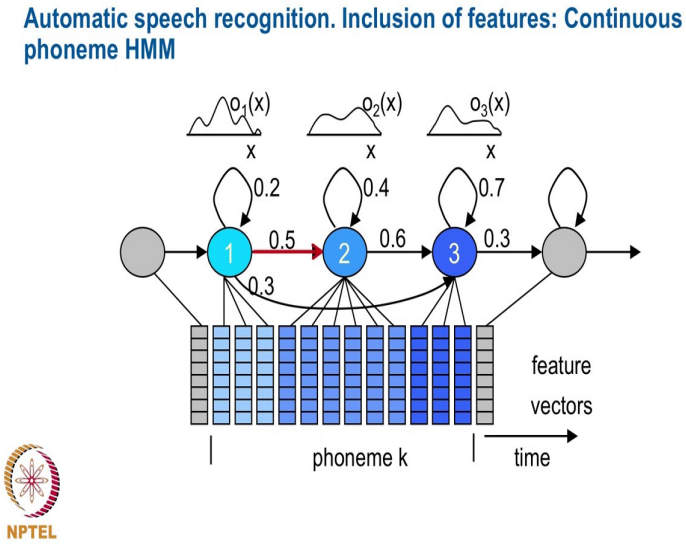
(Refer Slide Time: 15:49)



one; that is we have probabilities that this state 1 emits these vector features and this is a multidimensional vector.

We also have multidimensional, usually modelled in a mixed Gaussian way, ~~(16:02)~~ probabilities. And in addition to that we have of course the transition probabilities again.

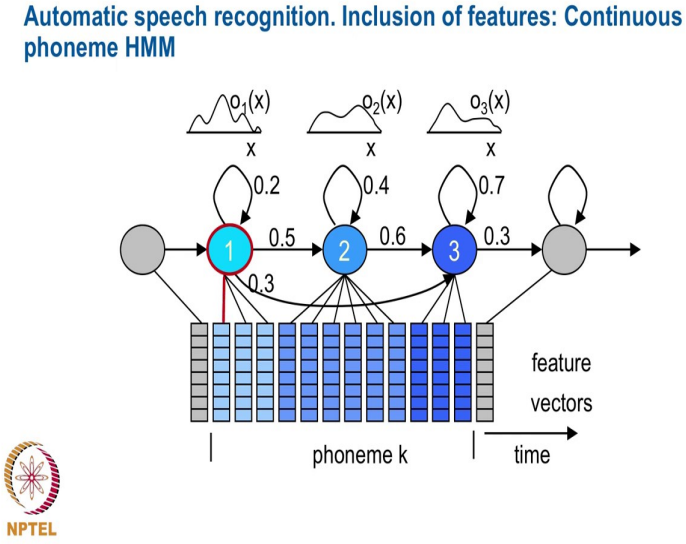
(Refer Slide Time: 16:09)



Now in order to calculate the probability that this sequence of feature vectors corresponds to a sequence of states which themselves are fixedly linked to a certain phoneme /k/ we have once again to multiply the probabilities.

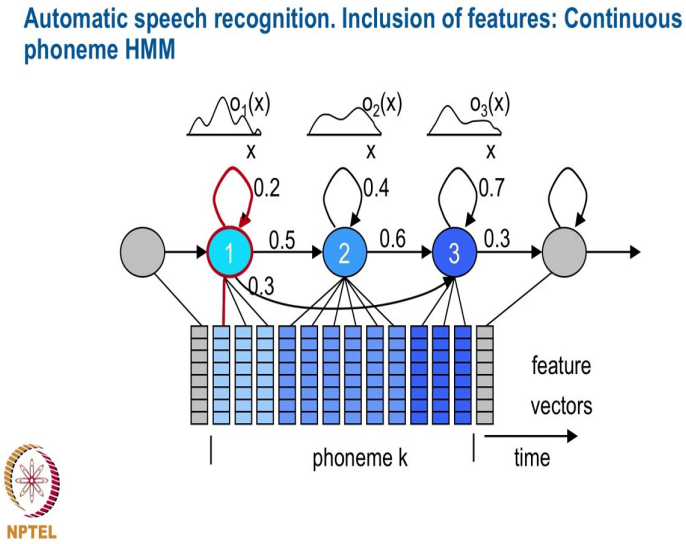
That is we have to multiply the probability that state number 1 ~~and its~~emits this feature vector

(Refer Slide Time: 16:31)



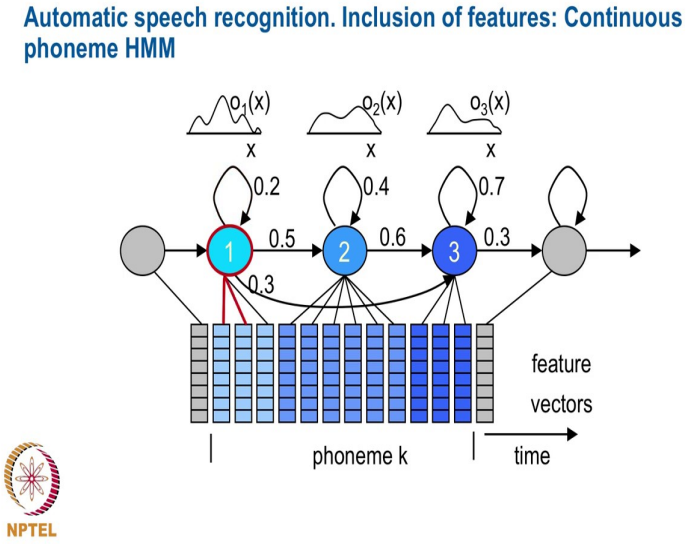
multiplied with the probability of staying in this state 0~~ point .~~2

(Refer Slide Time: 16:35)



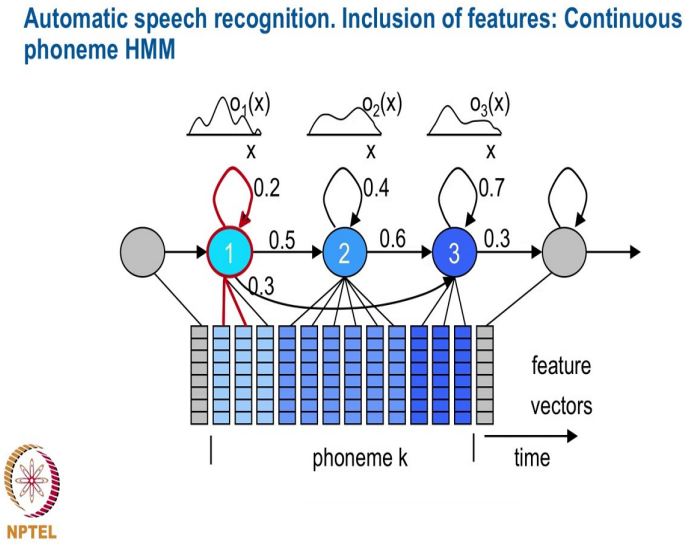
multiplied with the probability that the state number 1 emits this feature vector

(Refer Slide Time: 16:40)



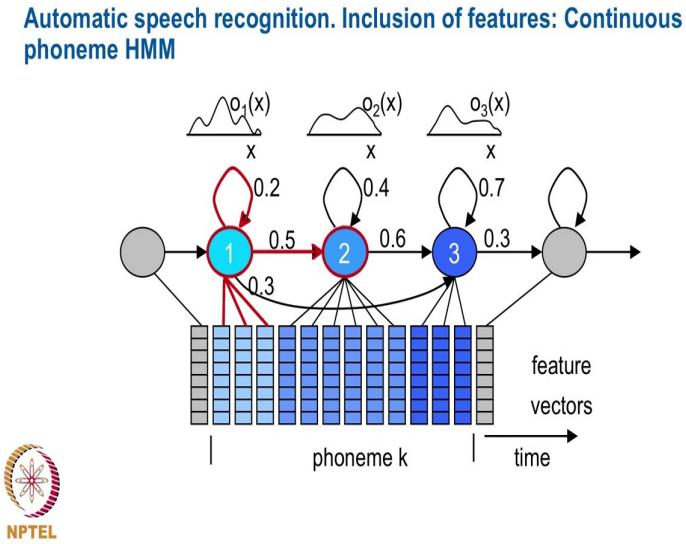
multiplied with the probability of staying in state 1 again

(Refer Slide Time: 16:44)



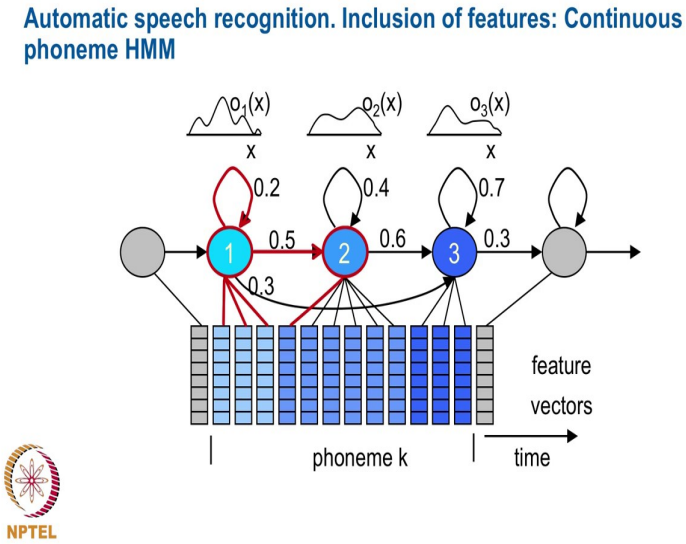
multiplied with probability that this state generates the third ~~type of (()) (16:47)~~light blue feature vector multiplied ~~by~~ with the probability to transit to state

(Refer Slide Time: 16:51)



number 2 multiplied with the probability that state number ~~emits~~ 2 emits this feature vector

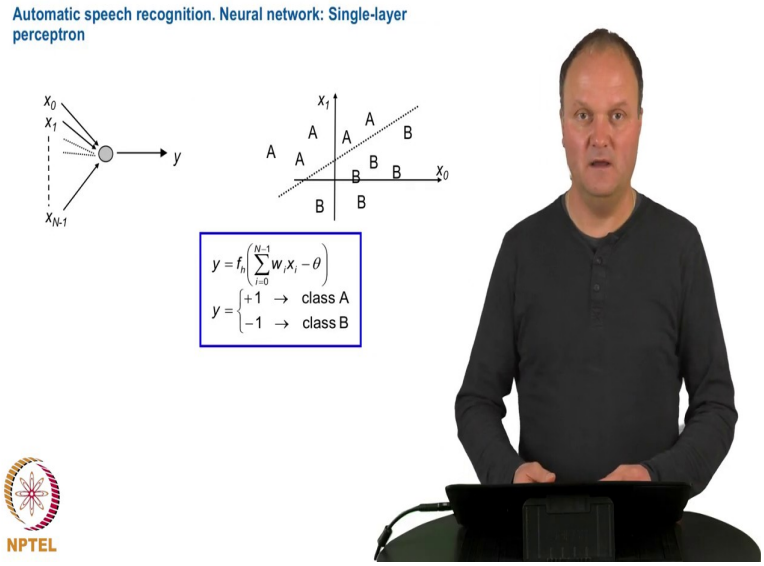
(Refer Slide Time: 16:56)



multiplied ~~by~~ with 0~~. point~~ 4 for staying in state 2 and so on.

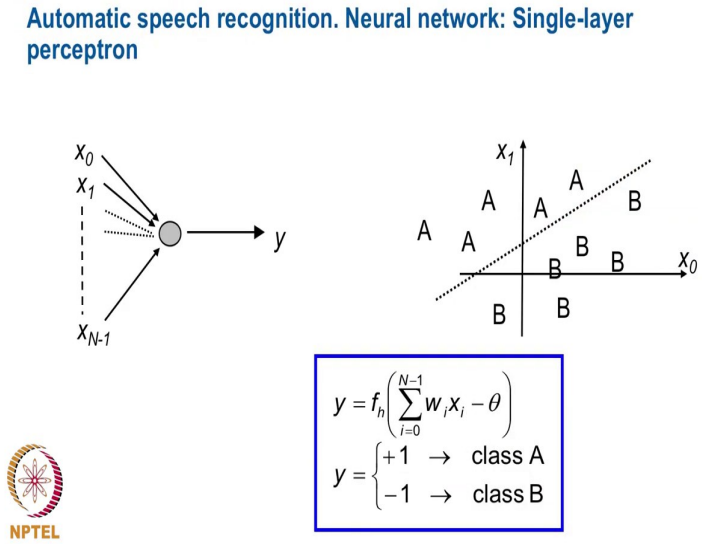
Once again we have a multiplication of all the probabilities in order to calculate the overall probability that ~~these~~ this sequence of feature vectors corresponds to the sequence of states which is fixedly linked to the phoneme ~~k~~/k/.

(Refer Slide Time: 17:15)



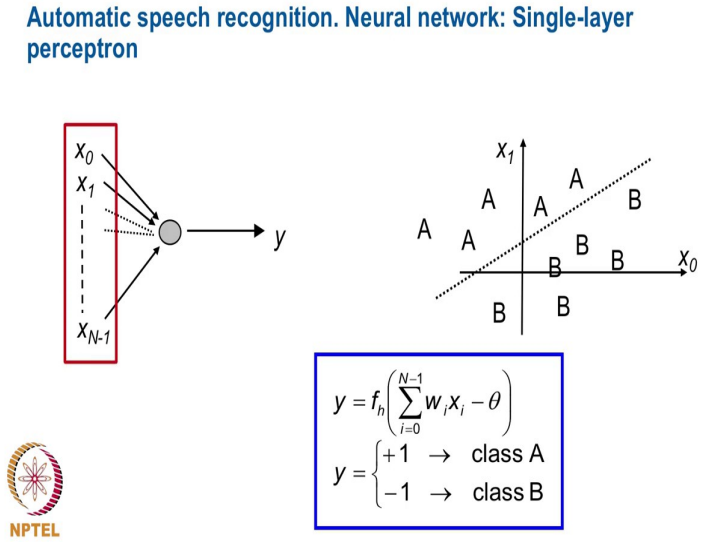
The second way of modeling the link between the feature vectors and the phonemes is ~~a~~the neural network. A neural network is in principle a very simple classifier and the most simple such classifier is shown here. It is called the perceptron.

(Refer Slide Time: 17:31)



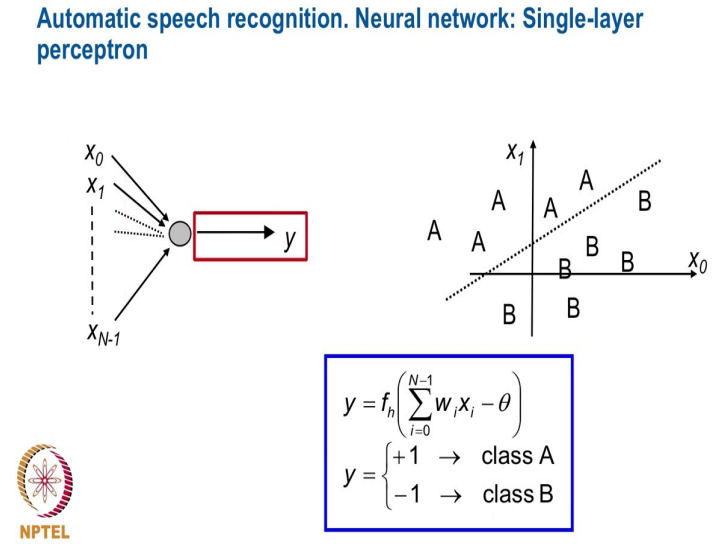
The perceptron takes a series of input values organized in a

(Refer Slide Time: 17:35)



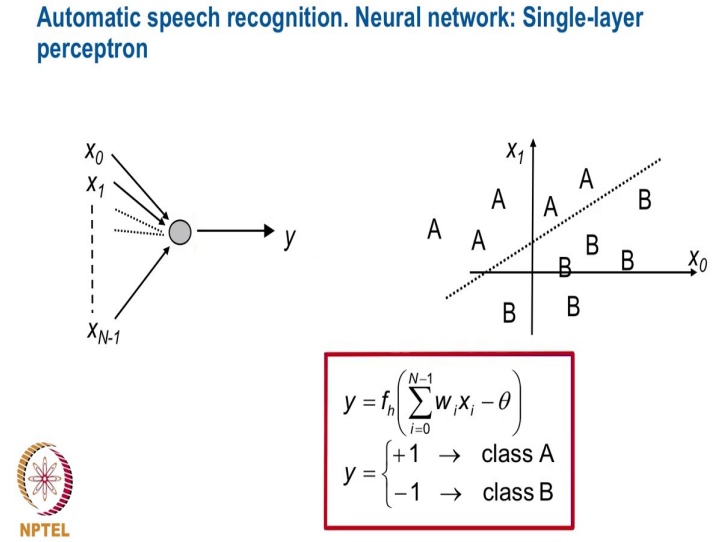
input features vector and gives a decision at the output. And the decision

(Refer Slide Time: 17:40)



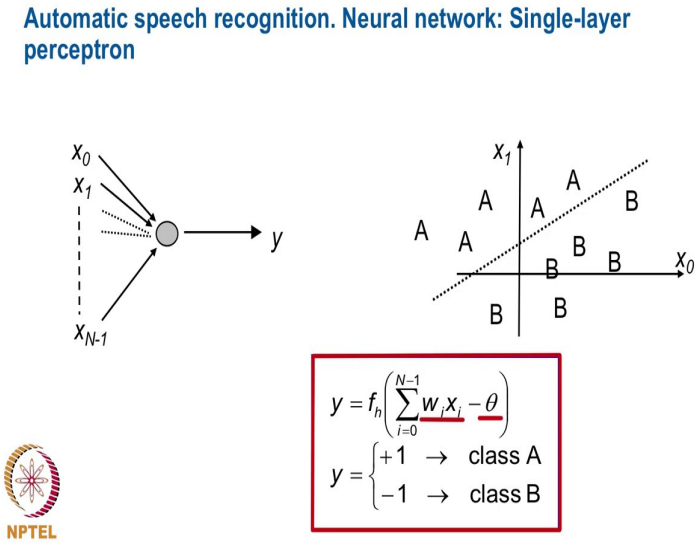
is calculated in thea~~a~~ way which we see here, that is we have a

(Refer Slide Time: 17:44)



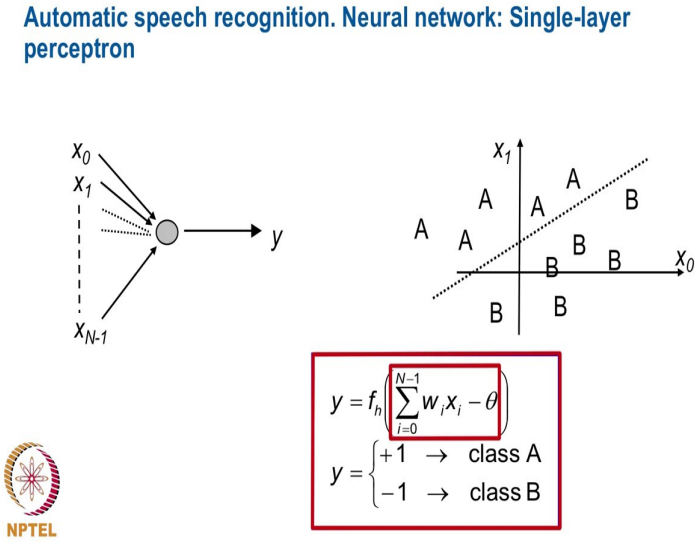
weighting of the individual components of the feature vector, subtract a bias here and then

(Refer Slide Time: 17:50)



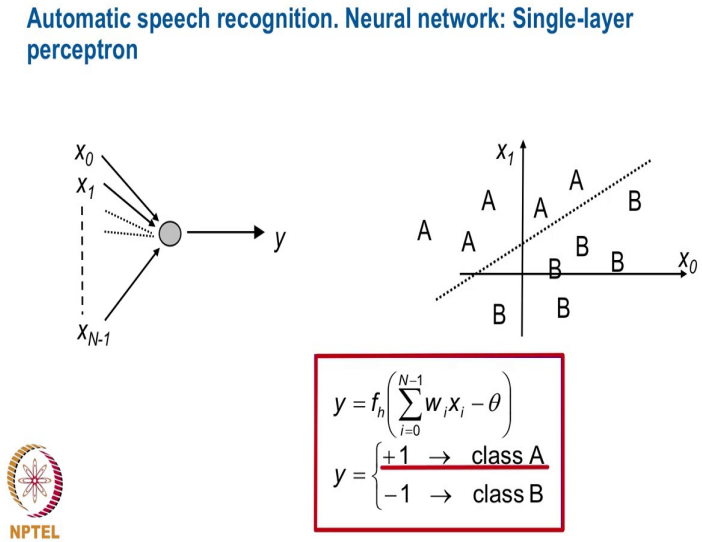
calculate the sum of all these.

(Refer Slide Time: 17:52)



And then apply a distinguishing function, threshold function which says that if we are above a certain threshold this feature vector belongs to class A and otherwise

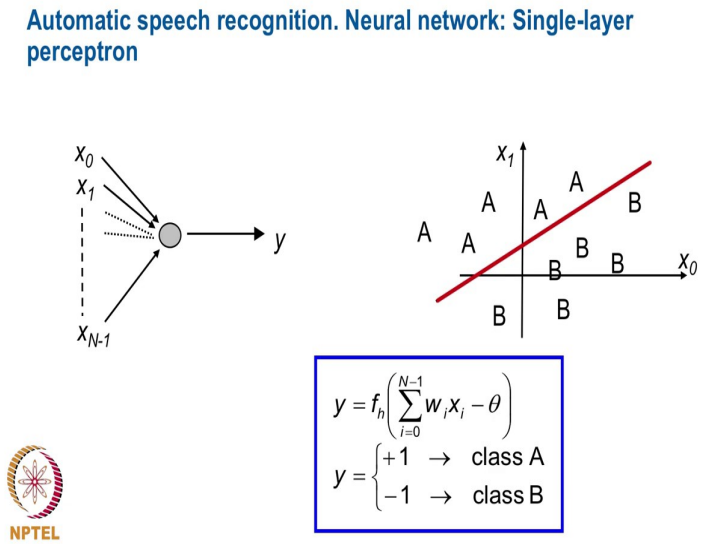
(Refer Slide Time: 18:04)



it belongs to class B.

In the two dimensional features space, you can easily imagine such a classifier to be a straight line for this

(Refer Slide Time: 18:13)

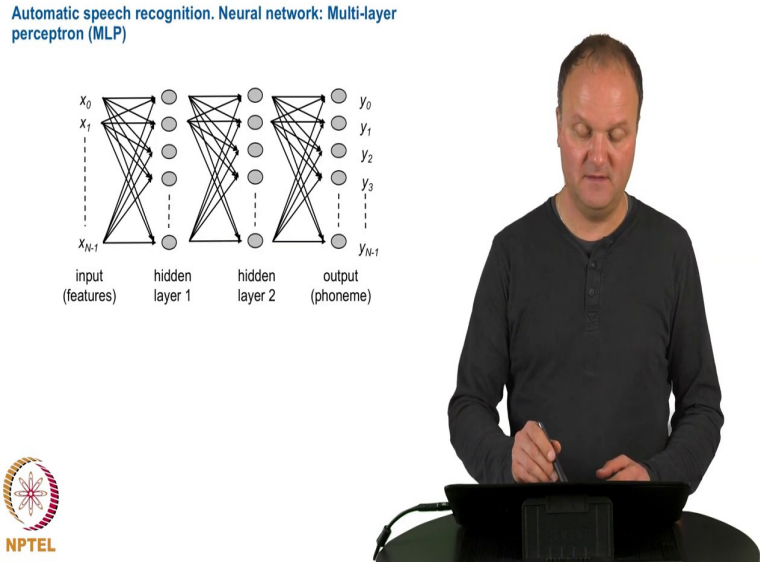


~~two dimensional~~two-dimensional space. If you go to higher dimensional spaces then you would have to imagine a plane in the three-~~-~~dimensional space or a hyper plane in an N-~~-N~~ dimensional general space. Usually feature vectors have a ~~high~~ higher dimensionality than 2 or 3.

Now this perceptron helps to distinguish between two classes but if you want to distinguish between all the sounds relevant for a certain language we need more than 2 classes. We have a couple of 25 to 30 different phoneme classes in different languages.

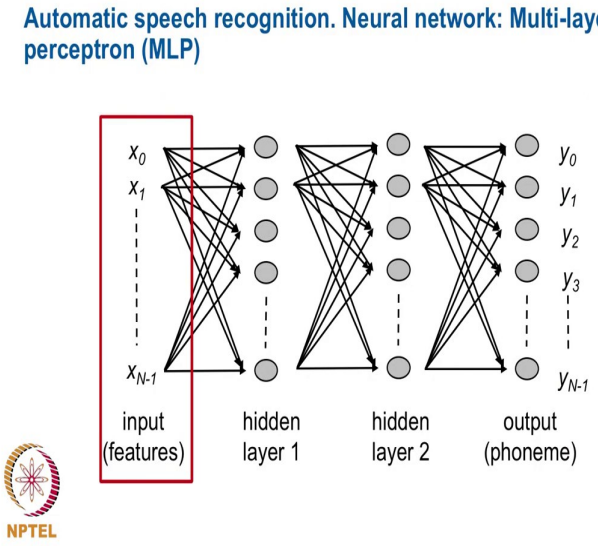
So we need to combine classifiers like this in order to create a type of network.

(Refer Slide Time: 18:55)



And this is shown here, and it is called a multilayer perceptron network or M-L-P network which combines these simple classifiers as you see here

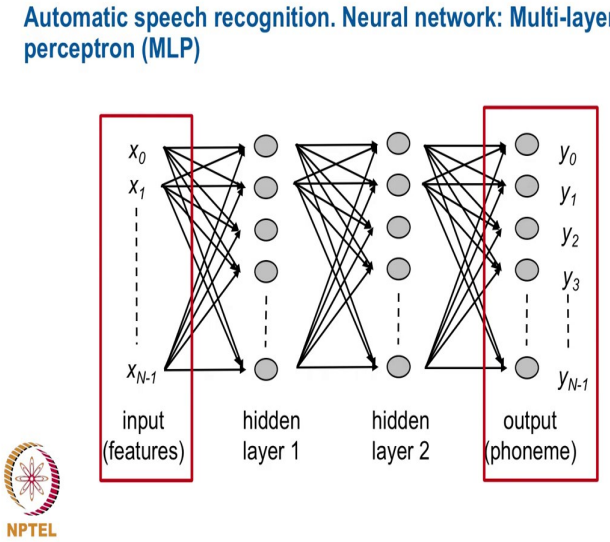
(Refer Slide Time: 19:06)



by taking the output of one classifier and using it as the~~an~~ input to another classifier and organizing these classifiers in terms of layers.

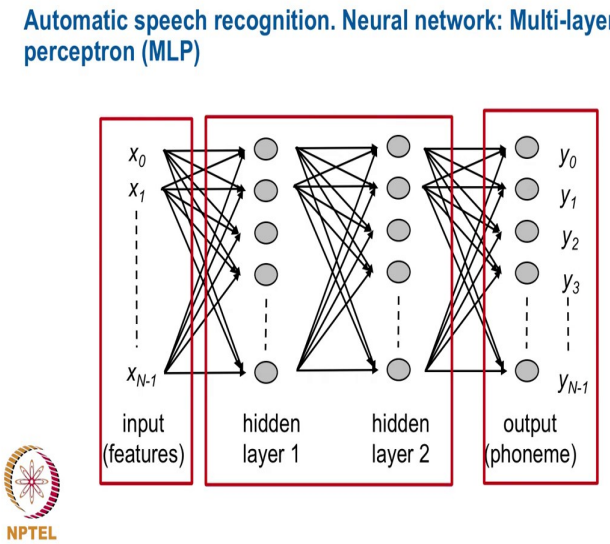
So usually you have an input layer and you have an output layer and you have

(Refer Slide Time: 19:19)



some hidden layers in-between.

(Refer Slide Time: 19:20)



In the input layers you have the input features; that is all the features of your feature vector, and in the output you have the different classes you would like to distinguish.

(Refer Slide Time: 19:47)

You do not necessarily need one output perceptron for each class but you can, for example combine them in a binary way so that you have 0 and 1 combinations in order to come up with the number of classes which you want to distinguish.

The last building block of our automatic speech recognizer is the language model. And the language model contains the information about the possible or probable sequences of words which then form sentences. There are two ways to formalize language models.

The first is in terms of a grammar. We usually work with context free grammars that are grammars where you have terminal symbols which are illustrated here and non-terminal symbols which are illustrated in terms of these brackets.

For example sentence can be, can have a positive statement or negative statement. A positive statement can be "yes" or "yeah" or "yes please". And the negative statement could be "no" or "no, thanks" or "no, thank you".

So the space of available options in these grammars remains relatively restricted. This is good for reducing the space of probabilities but in turn it only allows these sequences which are predefined by the grammar. So in order to make use of such a grammar in automatic speech recognition it needs to cover all possible combinations of words and sequences of words in order to be useful.

It also shows that you just cannot take a grammar from a grammar book but you would need to build your grammar in an application-specific way considering the possible sentences uttered by the user that they are grammatically correct or incorrect does not matter.

The second way of making use of sequential information is a statistical model. The statistical model defines the probabilities that certain words follow each other; that are probabilities for sequences of two or three words.

We then call it an n-grammar model~~. If it was two~~ and with two words it would be a bi-grammar model, with three words it would be tri-grammar model. The probabilities of the word alone would be the most simple case that is the unigram model.

These probabilities are determined by counting the frequencies of words in large corpora so ~~we~~ you need large corpora of reference text in order to calculate those probabilities.

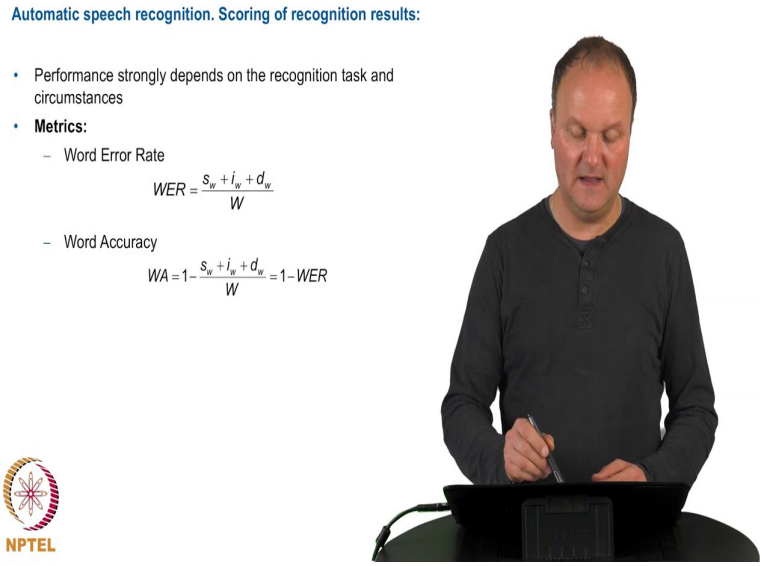
If a certain sequence of words doesn’~~ no~~t occur in that collection of text you could still calculate a kind of~~current~~ replacement probability by just multiplying the probabilities of the individual words that would be called a N~~N~~-gram backoff l~~L~~anguage model.

The probabilistic N-gram model is of course little bit more flexible because it allows you to generate sequences of words, so the probabilities and would not exclude in a fixed way other sequences of words which a grammar would exclude.

(Refer Slide Time: 22:37)

In order to calculate now how correct the output of such a speech recognizer is, that is an index of the performance of the speech recognizer, there are different metrics available. The most popular such metrics

(Refer Slide Time: 22:49)



are the word error rate and the word accuracy.

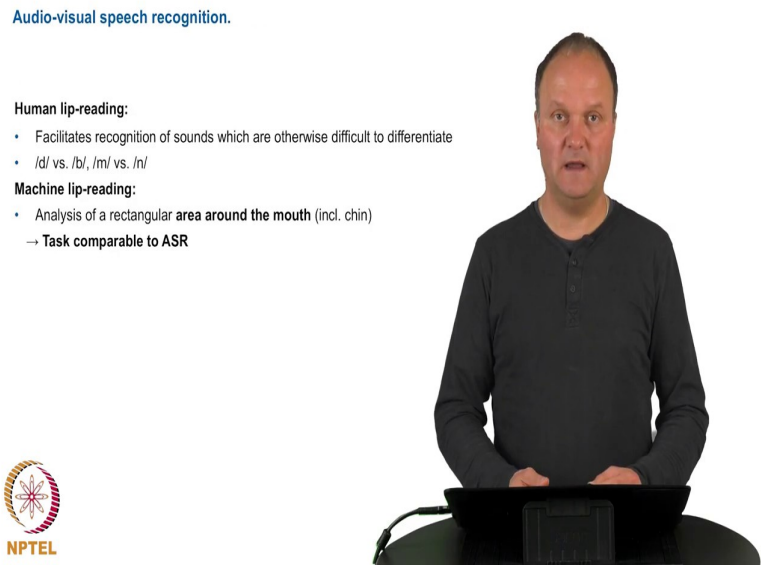
In order to calculate ~~the~~ a word error rate we first have to count the number of substituted words s_w ~~W~~, the number of inserted words by the speech recognizer i_w ~~W~~ and the number of ~~the~~ deleted words by the recognizer d_w ~~W~~ and divide that sum by the overall number of words in the reference transcription.

Now in order to determine whether ~~word~~ a word is really counted as a substitution, insertion or deletion, we need to first perform an alignment. And this is usually done through dynamic programming algorithms which punish the certain types of errors, substitutions, insertions and deletions in a certain way.

There are standardized way for, standardized methods for calculating this alignment and for the~~ir~~ determining a~~the~~ word error rate.

The word accuracy is actually 1 minus this word error rate for a individual word recognizer.

(Refer Slide Time: 23:50)



When I am speaking you might watch to my mouth on the video and this actually helps you to understand especially if there is background noise present.

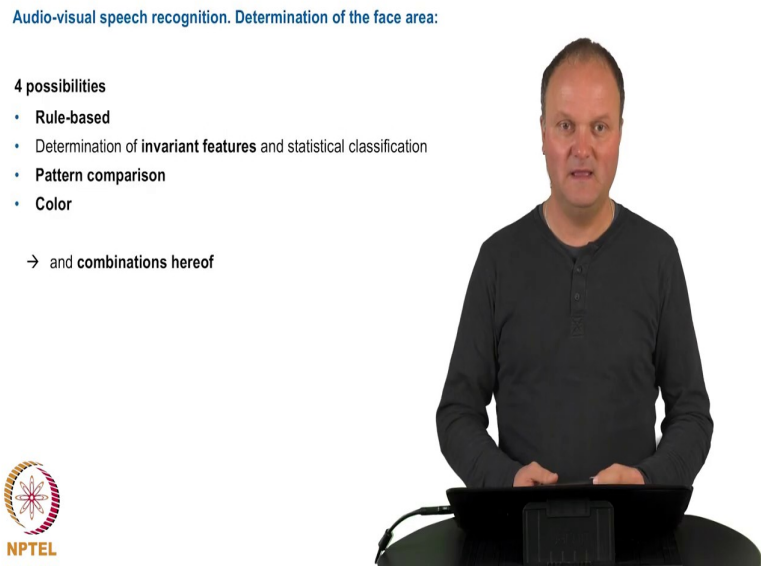
So humans do lip-reading and of course this can also be used, this is visual information, can also be used by machine in order to improve the recognition performance.

The lip-reading particularly facilitates the recognition of sounds which are otherwise difficult to differentiate or distinguish, like a /d/ from a /b/ or an /m/ from an /n/. Because of the shape of the mouth and lips particularly which are different in these pairs of sounds.

A m~~M~~achine can also do lip-reading, basically by analyzing the area around the mouth. In order to do so we first have to find the face, to track the face ~~and~~ then find the mouth in the

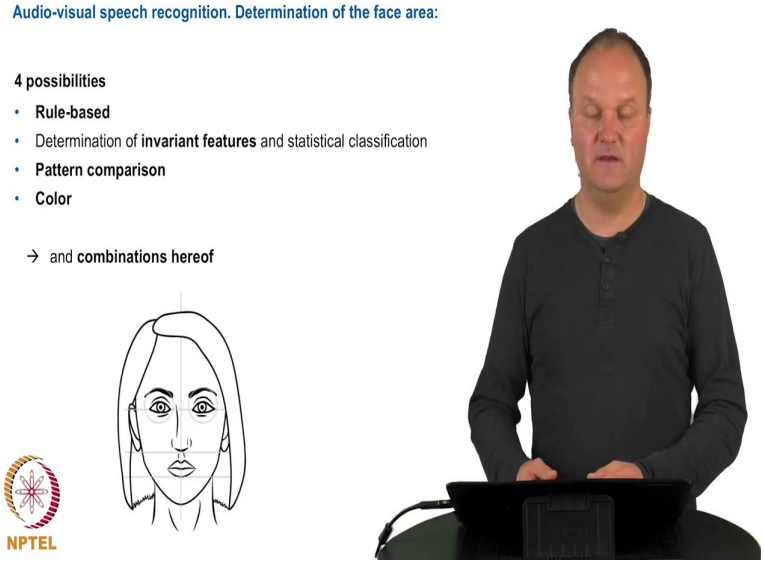
face region and then recognize ~~in the visual~~the individual shapes of the mouth in terms of so-called visemes that are the equivalence to phonemes on the visual level.

(Refer Slide Time: 24:56)



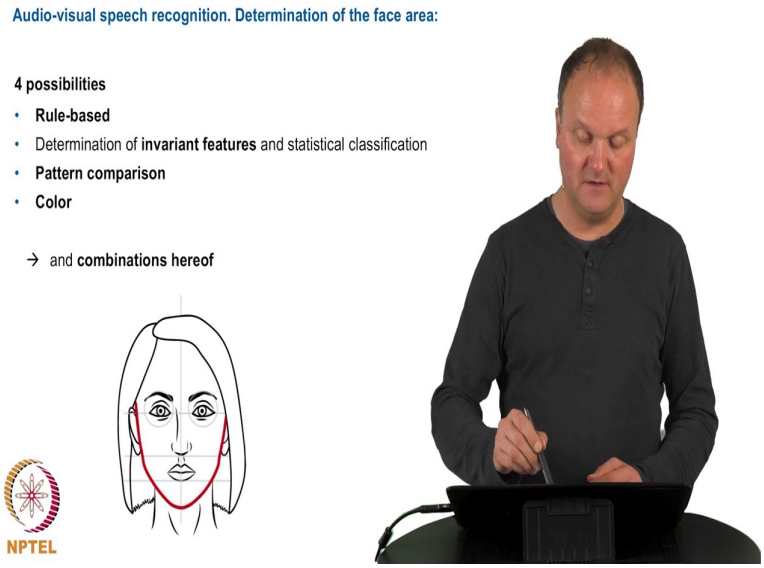
In order to find the face, there are 4 ~~4~~ possibilities. The first is a~~the~~ rule based approach,

(Refer Slide Time: 25:02)



we define the face like a combination of ovals

(Refer Slide Time: 25:05)



with certain points of interest like the eyes and the nose and the mouth in the face. This is like a child would depict a face through its most apparent characteristics.

This can also be defined as invariant features in a classical statistical classification approach. Then we can also do pattern recognition and we can also make use of the color.

Actually the color of the skin is something which is very unique and we can try to recognize the color in a image and then through the shape and the color, recognize the face in an image.

The color approach alone would of course also illustrate other parts of the body which show skin color, for example the hands and the arms would also be visible when you just use the color so in principle it is better to combine the different approaches in order to find and detect the face in an image.

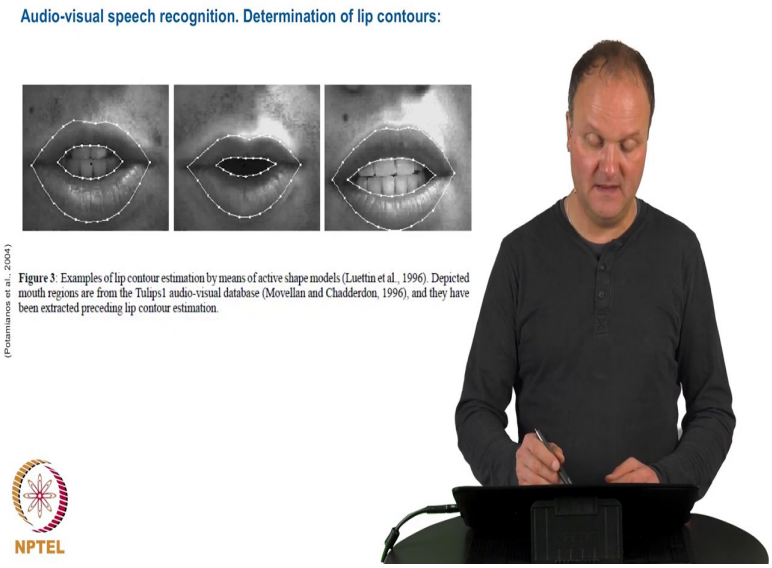
Here ~~is the~~you see some examples

(Refer Slide Time: 26:13)

of algorithms which have determined the face and then determined regions of interest in the face. Regions of interest are usually regions around the eyes and around the mouth and around the nose and on the front here.

So you see these points can be relatively well -identified in pictures despite the fact that these peoples illustrated in the pictures have different colors, lighting of the, of the pictures is not very good. But still it is possible to nicely find these points of interest and the regions of interest

(Refer Slide Time: 26:51)

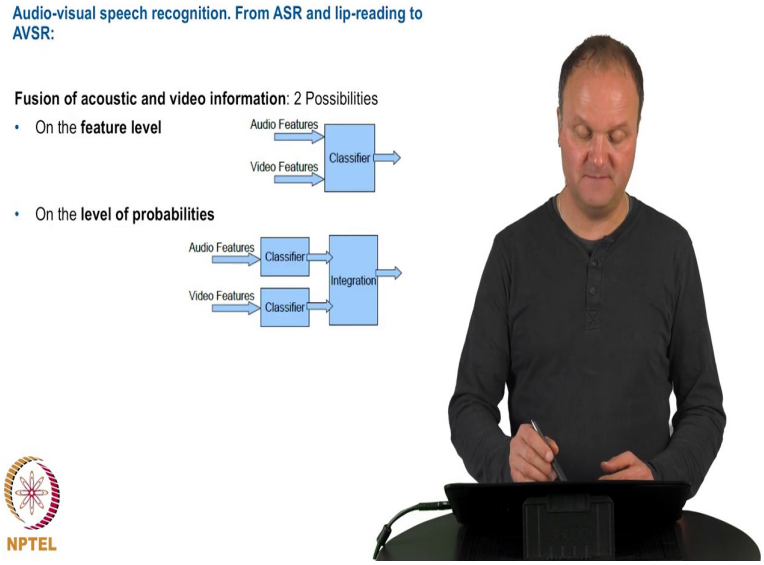


and then to go even further into detail for example to determine the contours of the lips.

And the contours of the lips help you a lot in differentiating the different sounds. Now this is the visual information part which needs to be translated into features and then you can follow the same approach as we have seen for the auditory features extracted for example, from a spectrogram or a Mel-scaled Cepstral Coefficients and so on.

There are two ways to combine the visual with the auditory information.

(Refer Slide Time: 27:23)



One is called the feature level fusion and the other is called the probability level fusion. In a feature level fusion you would take the audio features and the video features and just form a very long feature vector from both of them. And then do a standard classifying approach with the help of Hidden Markov Model or neural networks.

The second approach, the probability fusion is to train different classifiers, one for the audio features representing phoneme classes and then the video features representing viseme classes and then combining these at the decision level or the level of probabilities into one unique sequence of recognized words.

Both approaches are in principle feasible if you deal with audiovisual speech recognition what is the term for this type of speech recognizer.