5. Automatic speech recognition (ASR)

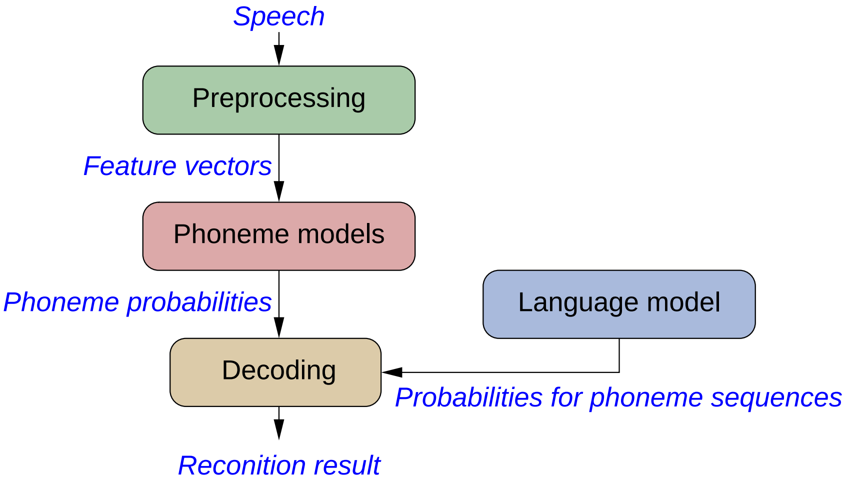
By **automatic speech regonizer** we mean a system that interprets human speech in some way. Possible applications include: speech-controlled devices, dictation, automatic transcription of meetings, automated telephone services, searching large audio/video archives, speech translation.

The performance of current speech recognition systems can be anything between perfect and poor depending on the following factors:

- Vocabulary size (only a few words ... unlimited vocabulary)
- Speech style (isolated words ... natural speech)
- Language style (formal speech ... spontaneous conversation)
- Speaker dependency (trained for one speaker ... speaker independent)
- Recording conditions (quiet room ... noisy restaurant)

Speech recognizer

A typical speech recognizer consists of the following modules. Additionally, a complete application may require methods for separating speech and non-speech, dialogue control, speaker recognition, and search methods.



Speech

Preprocessing

Feature vectors

Phoneme models

Phoneme probabilities

Language model

Decoding

Probabilities for phoneme sequences

Reconition result

How does it work?

When building a recognizer, we train two models:

- **Acoustic phoneme model**: Computes phoneme probabilities in given speech signal.
- **Language model**: Defines legal phoneme sequences and their probabilities.

In principle, after the models have been trained, recognizing a new speech signal is simple: Find the phoneme sequence that is best according to the phoneme model and language model.

Demonstrations: small vocabulary, large vocabulary.

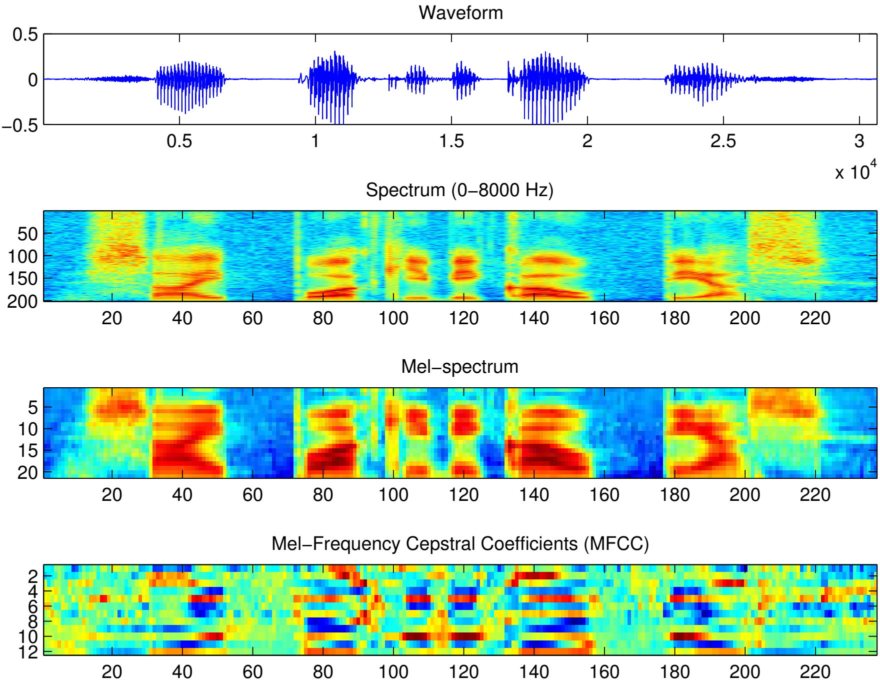
Lessons learnt from the demo:

- Recognition with a small vocabulary is easy. Especially, if the words are different from each other acoustically.
- Simple vocabulary-based recognition does not work well if the vocabulary is not restricted. Especially in Finnish, inflections and compound words worsen the problem.

Solutions to this problem presented later during the course (lectures: statistical language models, methods for speech recognition)

Preprocessing the speech signal

The preprocessing module extracts relevant features from the speech signal. Typically features are computed from the short-time fourier spectrum.



Waveform

Spectrum (0−8000 Hz)

Mel−spectrum

Mel−Frequency Cepstral Coefficients (MFCC)

Acoustic phoneme models

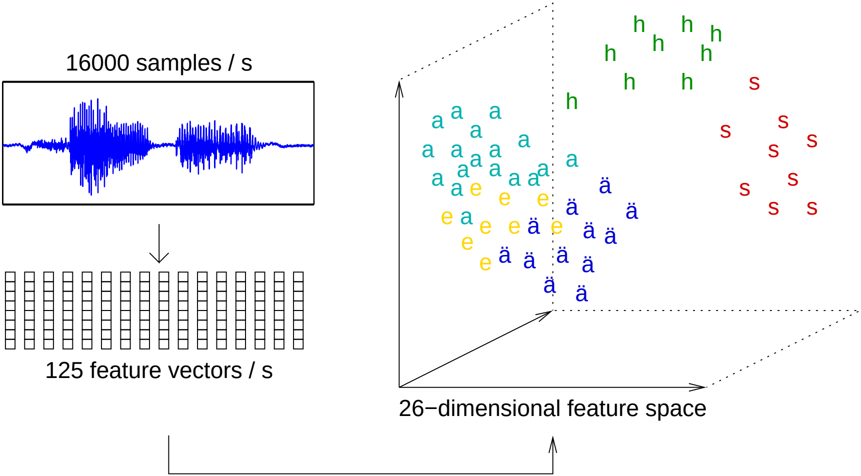
For training the phoneme models using Hidden Markov Models (HMM), tens or hundreds of hours of transcribed speech is needed.

- For each phoneme, fetch feature vectors of all speech segments that contain the phoneme. The feature vectors form a cloud of points in the feature vector space which typically has 26-39 dimensions.
- Model the cloud with multi-dimensional gaussian mixtures.

The simple model above can be improved:

- Divide each phoneme in three states. The idea is that the first state models the transition from the previous phoneme, and the third state the transition to the next phoneme.
- Create context-dependent models for each phoneme: For example, /a/ between /p/ and /u/, /a/ between /k/ and /t/, and so on.

The feature vector space illustrated

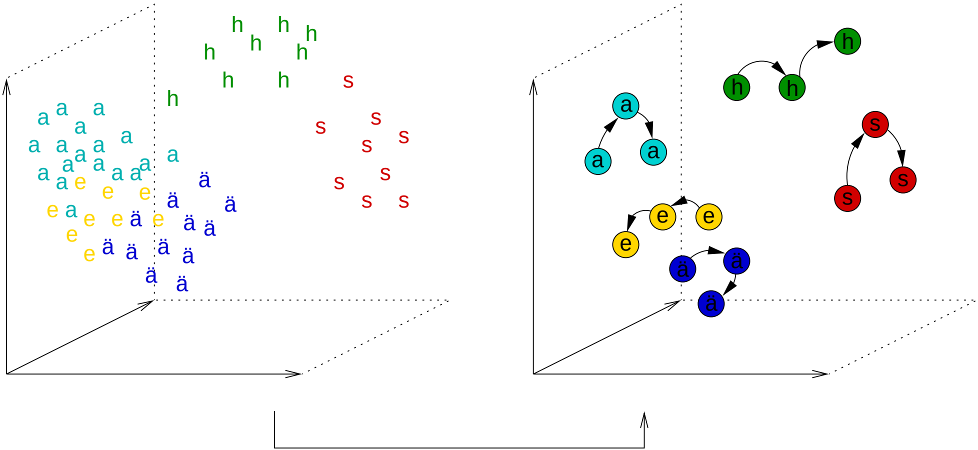

16000 samples / s

125 feature vectors / s

26−dimensional feature space

Perfect features would separate different phonemes in the feature space. In practice, the phoneme clusters overlap, unfortunately.

Phoneme models in the feature space



With Hidden Markov Models one can efficiently compute the state sequence that best matches the given speech signal. More about Hidden Markov Models later during the course.

Language model

Basically, the **language model** defines what kind of sentences or commands the recognition expects from the user.

- Simple command interface: list of commands, all equally probable.
- Telephone service: A state machine describing the phases in a dialogue, and expected responsees in each state.
- Dictation: Statistical n-gram model trained from a large collection text documents.

Language model limits the search space (possible hypotheses) and helps to rank acoustically similar hypotheses: *"Give those papers to me, please."* versus *"Give toes pay purse two me, police."*

Demonstration: Speech recognition using a good n-gram language model.

Some recent recognition results

Finnish (TKK, 2005):

Task	Speaker dependent	WER
Audio book	yes	7
Radio news	no	22
TV news	no	35
TV debate	no	70

English (HTK-system, University of Cambridge)

Task	Speed	Year	WER
News (NIST)	10	2004	11
News (NIST)	1	2004	15
Telephone (SWB)	> 100	2005	24
Telephone (SWB)	< 10	2005	27

WER = word error rate (%)
Speed = recognition speed (x Real-time)

Application: Spoken document retrieval

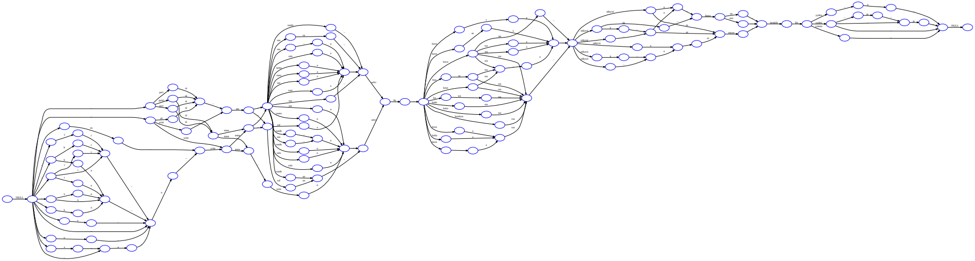
The user makes queries to a large speech database, and the system tries to find the most relevant clips. For example, broadcast companies have massive audio and video archives that are often untranscribed.

Speech recognition system can be used for transcribing the archive automatically. Relevant clips can be found even if the word error rate in recognition more than 20 %.

Demonstration: http://speechfind.utdallas.edu/

Word graphs or lattices

In some applications, it is useful to get multiple recognition hypotheses or a so called *word graph* or *lattice*.



Word graphs can be used for estimating how confident the regocnizer is about the recognition output.

Also, in spoken document retrieval, word graphs are useful: A relevant query word may be found in the word graph even if it would is incorrectly recognized in the best hypothesis.