Loria
Laboratoire lorrain de recherche en informatique et ses applications

Speech modeling and Automatic speech recognition

Romain Serizel (Slides by D. Jouvet)
LORIA – INRIA - Nancy

Inria informatics mathematics
cnrs
UNIVERSITÉ DE LORRAINE

Loria

Laboratoire lorrain de recherche
en informatique et ses applications

Introduction

Information carried by speech

- Linguistic content (words)
  - Speech recognition
    - Recognition of all uttered words, or just some keywords

    ➔ Vocal commands, speech transcription, vocal indexing, …
- Speaker (who speaks)
  - Speaker recognition
    - Speaker identification, or speaker authentication

    ➔ Diarization (associating speech segments with speakers), …
- Language
  - Language recognition
    - Identification of the spoken language, or of the dialect, accent, …
- Paralinguistic information
  - Emotions
    - Neutral speech, joy, sadness, anger, …
  - Speaking style
    - Spontaneous vs. read speech, sport commentary, …

Automatic speech recognition (ASR)

- In the seventies, rule-based approaches aiming at
  - Splitting the speech signal into sounds
  - And, identifying phonemes, then words, then sentences

- Since a few decades, data-based learning approaches have spread (data correspond to examples of words or sentences)
  - Acoustic templates
    - Recording of a few pronunciations of words that are used as references

    ➔ Recognition relies on comparing an unknown acoustic form to known reference templates using dynamic programming (DTW: Dynamic Time Warping)
  - Hidden Markov models (HMM)
    - Statistical models representing the pronunciation of a phoneme or of a word

    ➔ Training model parameters, recognition process, structures, adaptation, …
  - Neural networks
    - Currently the most efficient and most often used approach
    - Take benefit of large data sets and computation resources

    ➔ Application to speech recognition, structures, …

Examples of automatic speech recognition applications

- Vocal commands (for environment, voice mail, customer services, …)
- Vocal dictation
- Automatic transcription of speech data
- Vocal assistants
  - Google assistant, Siri (Apple), Cortana (Microsoft), Alexa (Amazon)
  - Used on smartphones, or through home devices

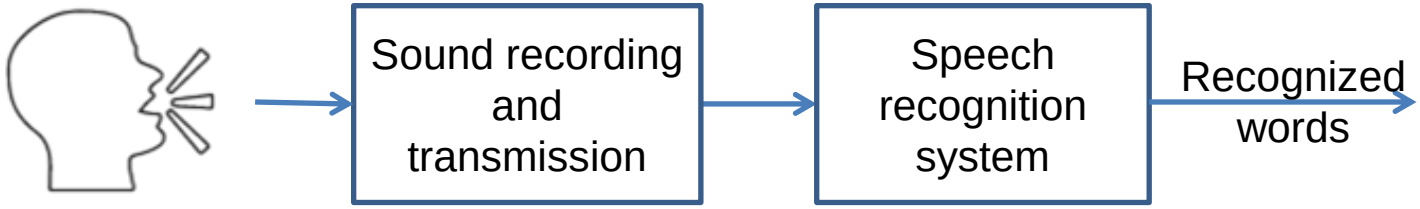
Loria

Laboratoire lorrain de recherche
en informatique et ses applications

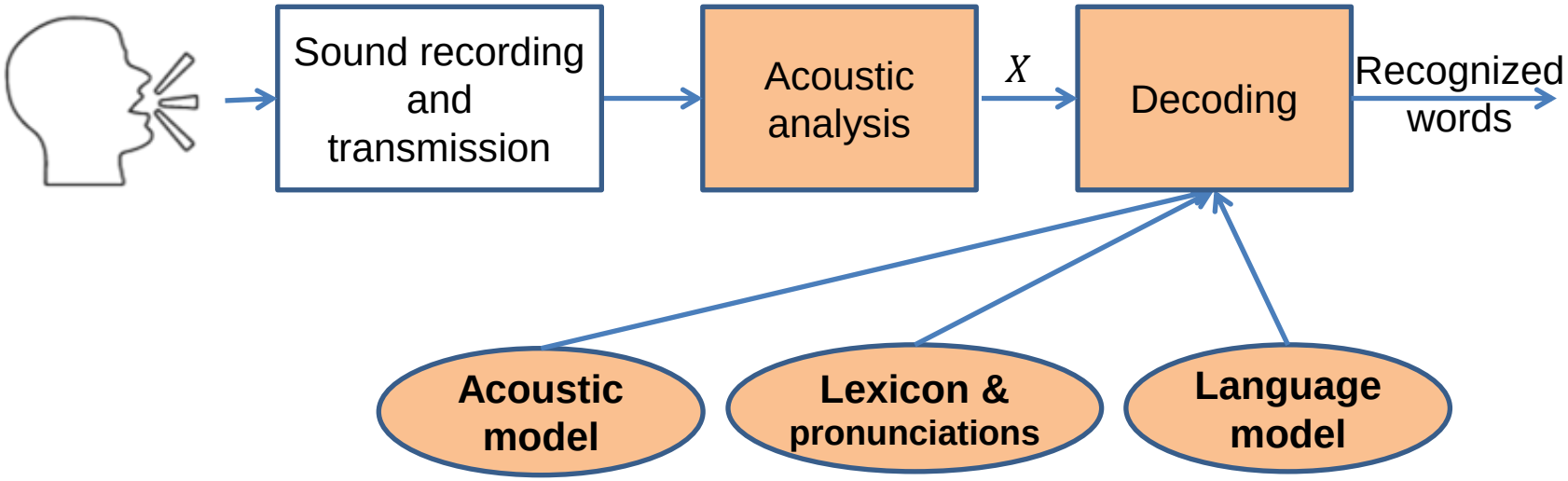
Automatic speech recognition

Automatic speech recognition (ASR)

Consists in identifying the words that are pronounced



Automatic speech recognition

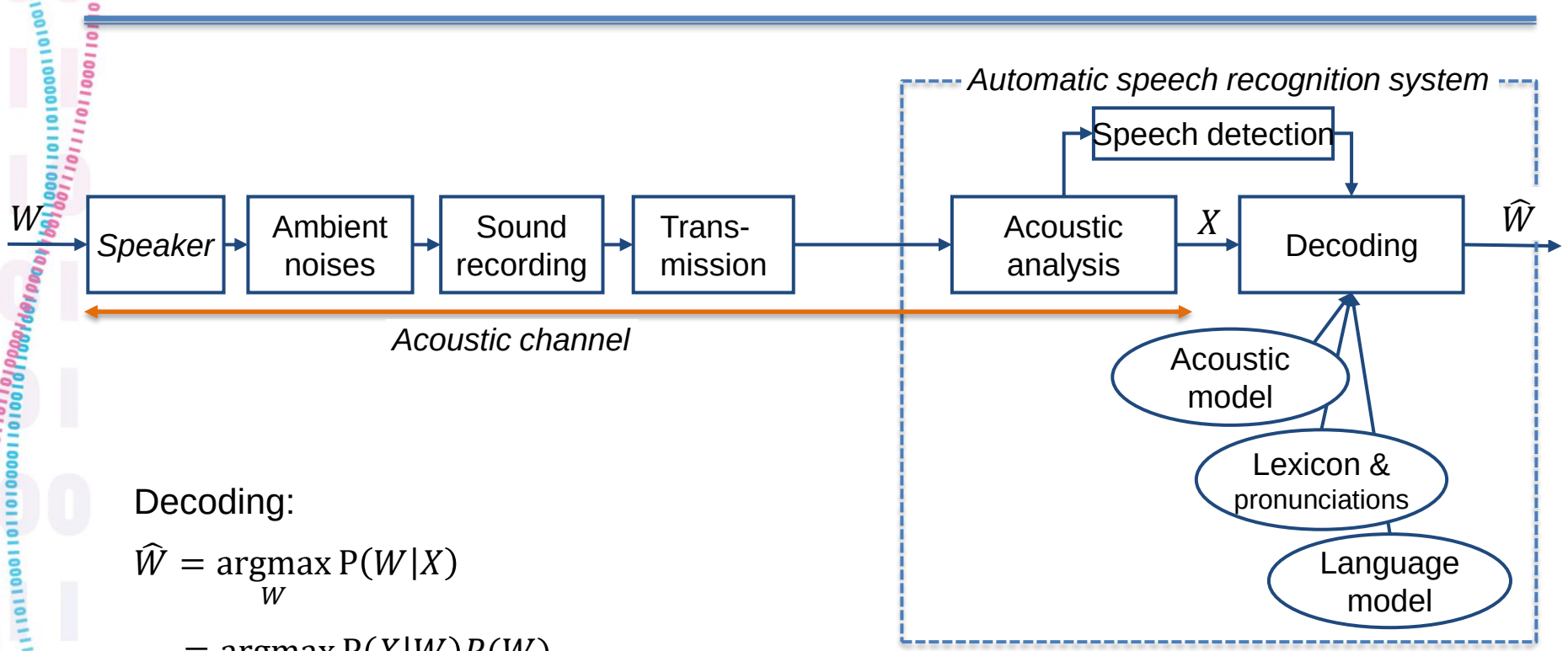


Represents how sounds are pronounced (for one or several speakers, in a given environment)

List of the words known by the system, with their pronunciation variants

Possible sequences of words

Automatic speech recognition - decoding



Decoding:

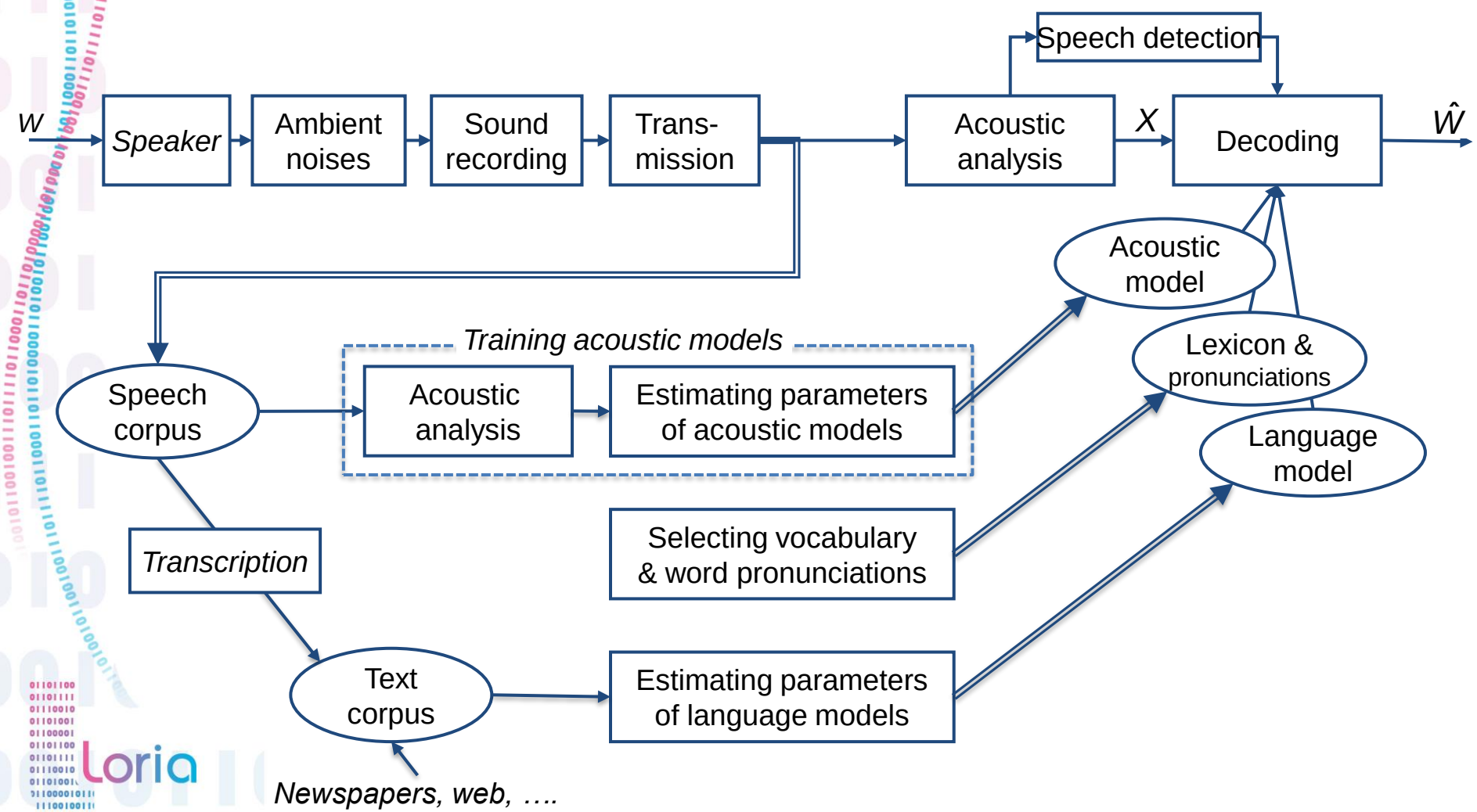
$$\widehat{W} = \underset{W}{\operatorname{argmax}}\, \mathrm{P}(W|X)$$

$$= \underset{W}{\operatorname{argmax}}\, \mathrm{P}(X|W)P(W)$$

Acoustic model & word pronunciations

Syntax

Automatic speech recognition - modeling



Content

- Speech
  - Production, perception, sounds of the language, …
- Basics of speech recognition
  - Acoustic analysis
  - Acoustic templates
  - Hidden Markov models
- Automatic speech recognition
  - Lexicons, language models, continuous speech recognition
  - Variability, performances, robustness & adaptation
- Deep neural networks
  - Neural networks, reccurent networks, convolutional networks, ...
  - Application to acoustic modeling and language modeling
- Extracting other information
  - Para-linguistic information
  - Speaker and language recognition

Speech

Basics of speech recognition
Automatic speech recognition
Deep neural networks
Extracting other information

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Speech

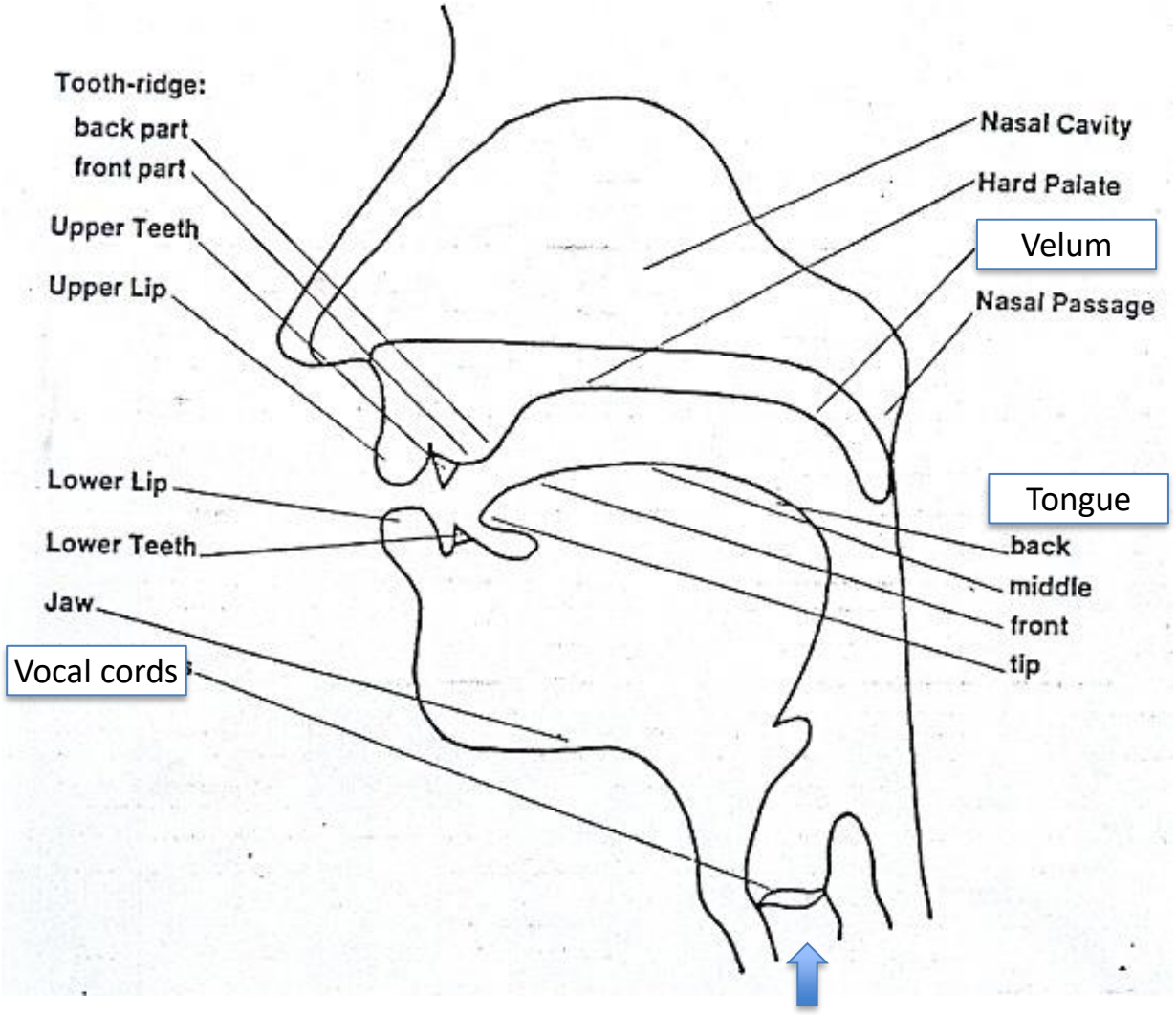
Speech

- **Speech production & perception**
  - Spectral analysis (time-frequency)
  - Fundamental Frequency & formants
  - Sounds of the language
  - Speech signal variability
- Basics of speech recognition
- Automatic speech recognition
- Deep neural networks
- Extracting other information

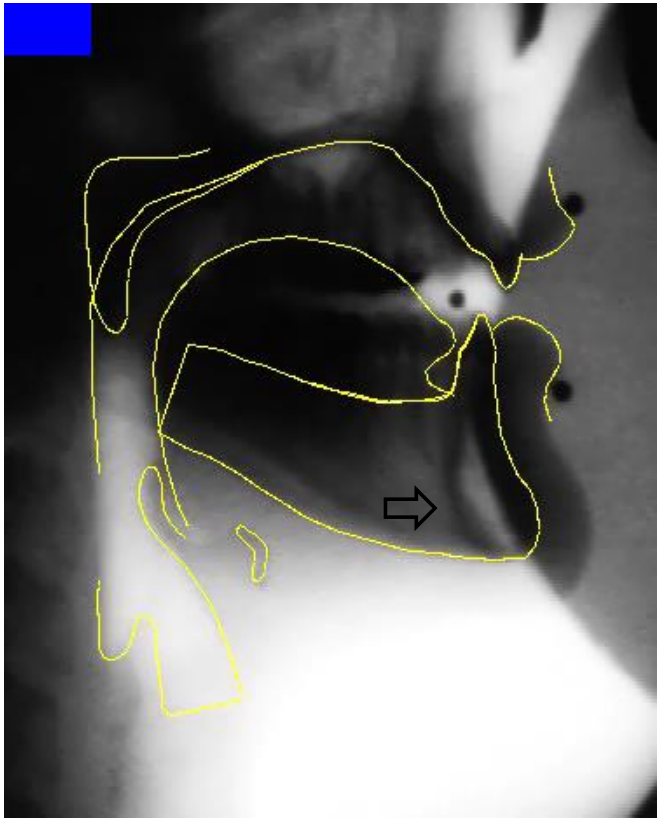
Loria
Laboratoire lorrain de recherche en informatique et ses applications

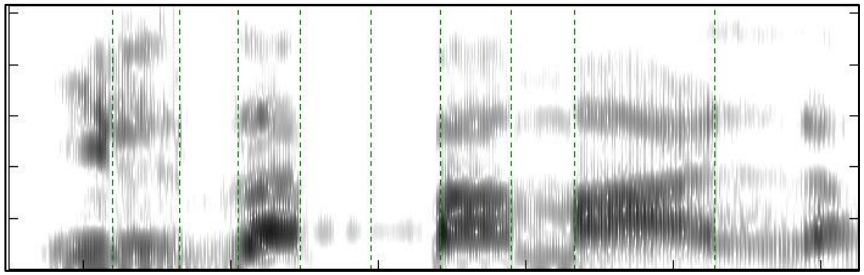
Speech production & perception

Vocal tract


Tooth-ridge:
back part
front part
Upper Teeth
Upper Lip
Lower Lip
Lower Teeth
Jaw
Vocal cords
Nasal Cavity
Hard Palate
Velum
Nasal Passage
Tongue
back
middle
front
tip
Air from the lungs

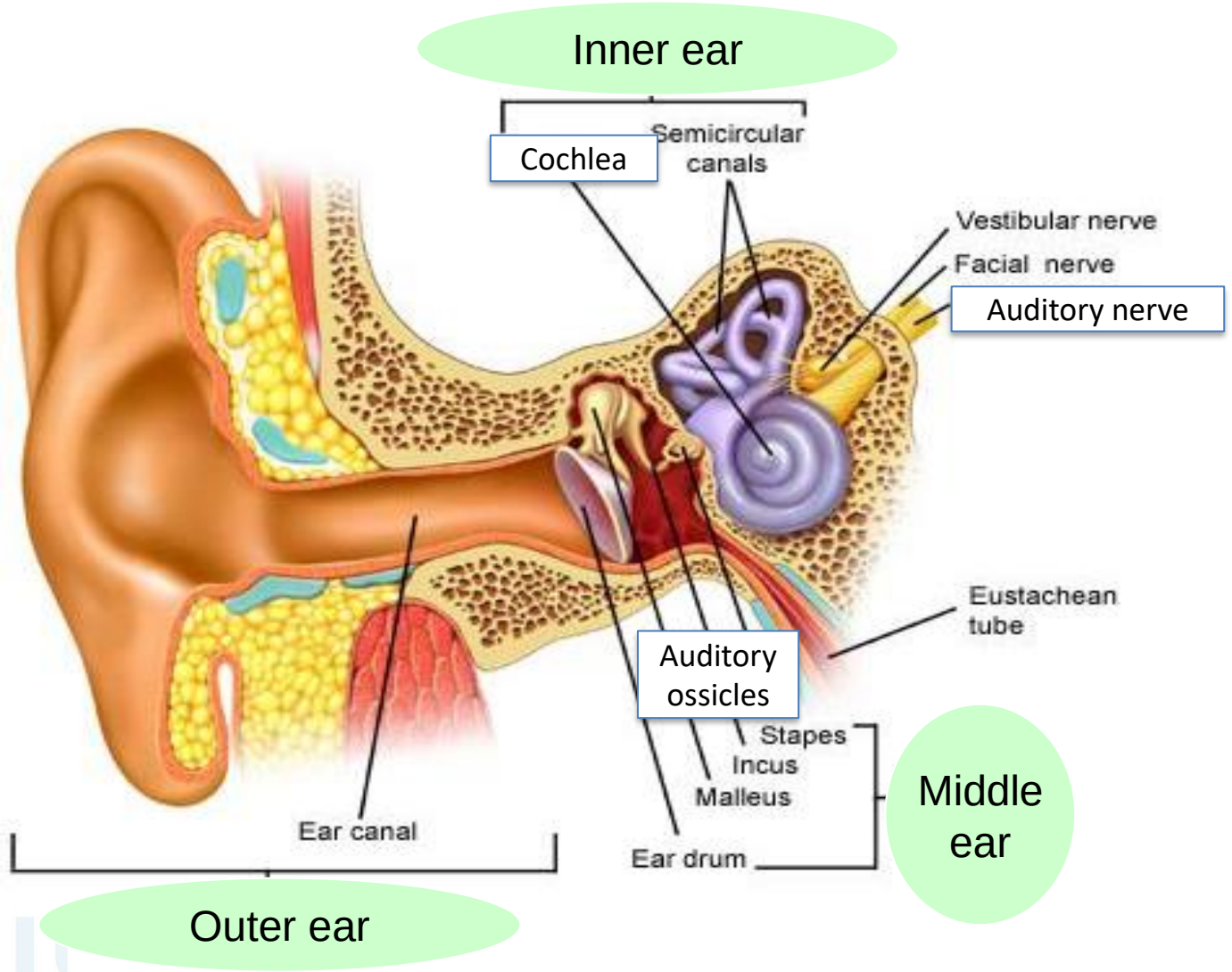
Speech production (French example)





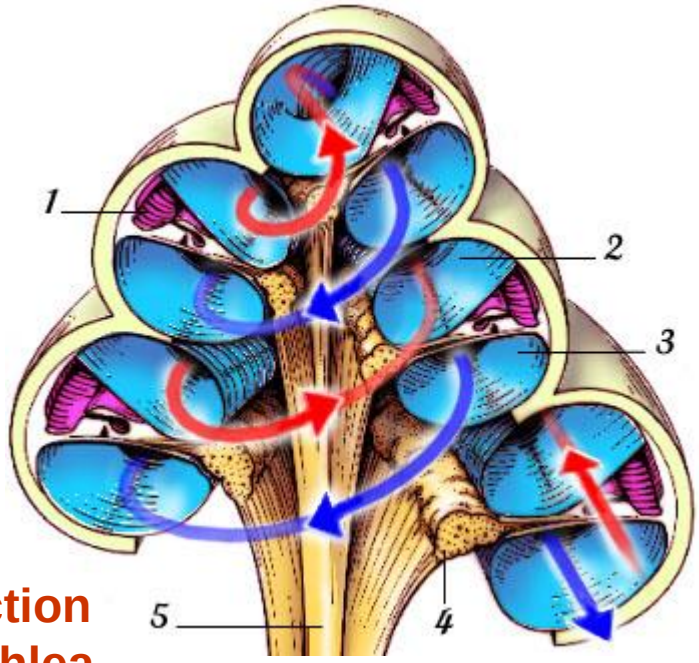
Il zappe pas mal.

Anatomy of the ear



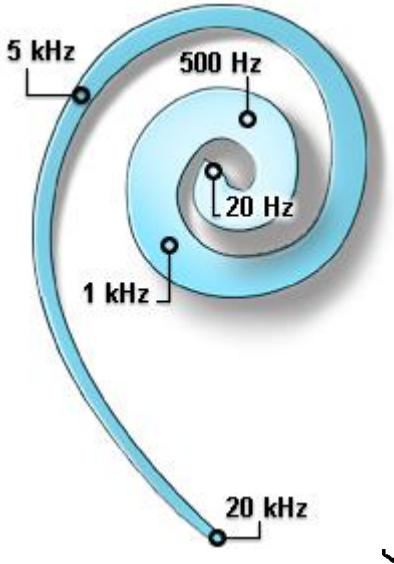
Hearing organ - the cochlea

Cochlea : the cochlea's hollow tube is about 32 mm long and 2 mm in diameter, wrapped like the shell of a snail (about 2.5 times)



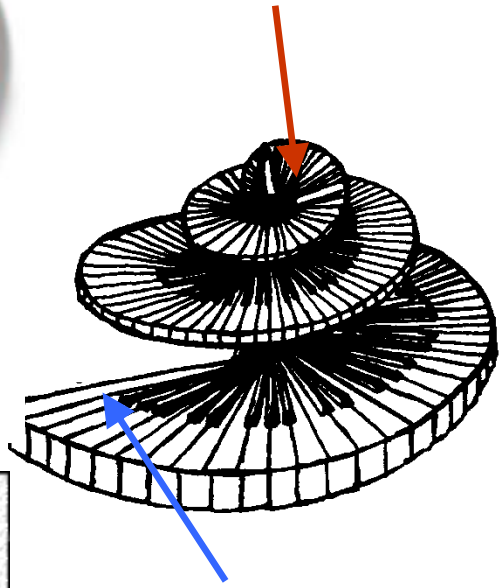
Cross section of the cochlea

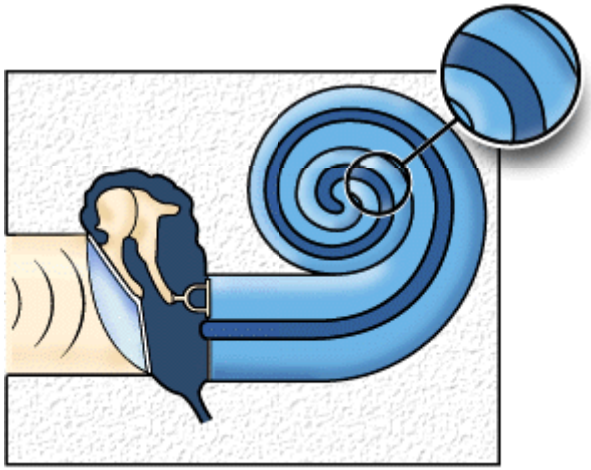
Tonotopy in the auditory system

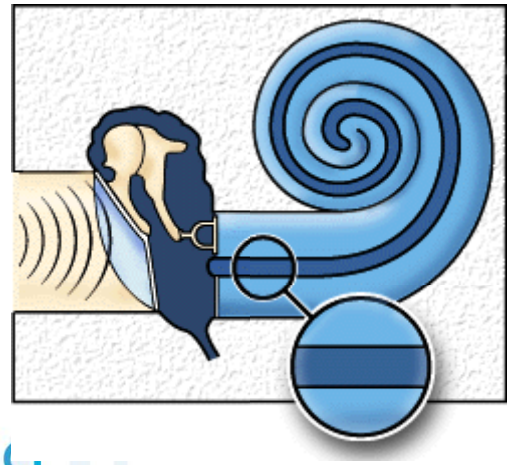


Basilar membrane

apex → low frequencies







base → high frequencies

Tonotopy → spatial arrangement of where sounds of different frequency are processed

- high frequencies are processed near the base
- low frequencies are processed near the apex

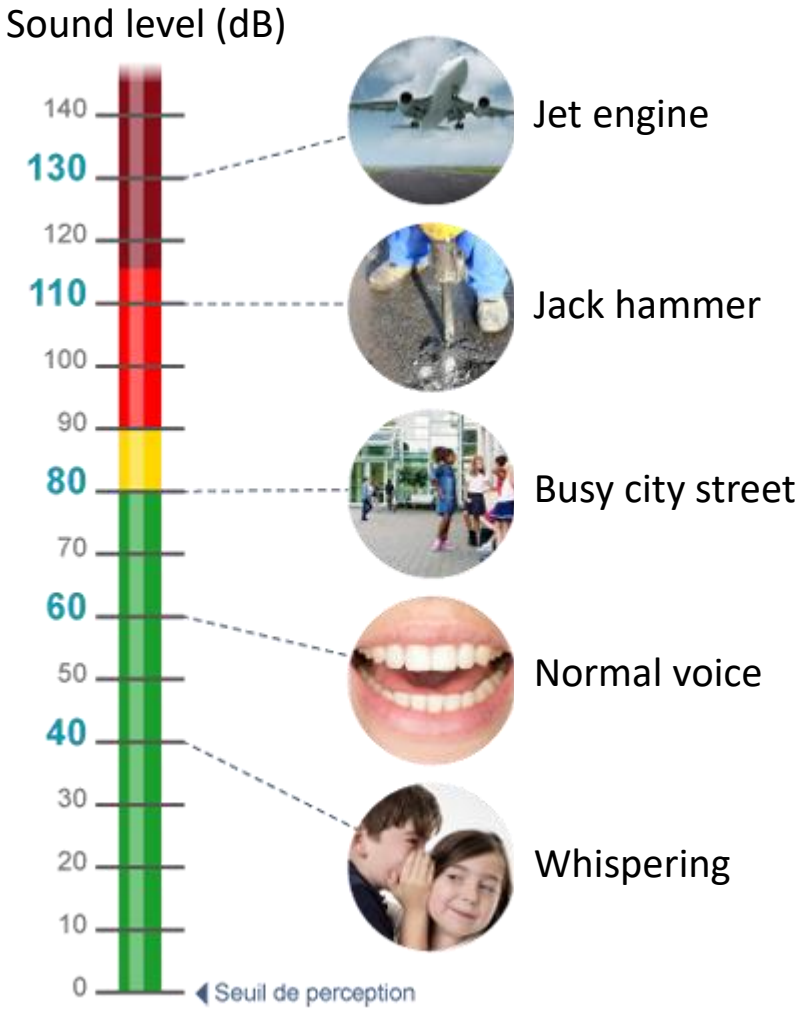
Auditory perception

- Human hear sounds between 30 Hz and 20 kHz
  limits vary with age, hearing disorders, etc.
- With age, there is a progressive loss of the ability to hear high frequencies
- Speech sounds frequencies are mainly below 8 kHz
- Even when considering only frequencies below 4 kHz, the speech signal is intelligible (e.g., telephone speech)

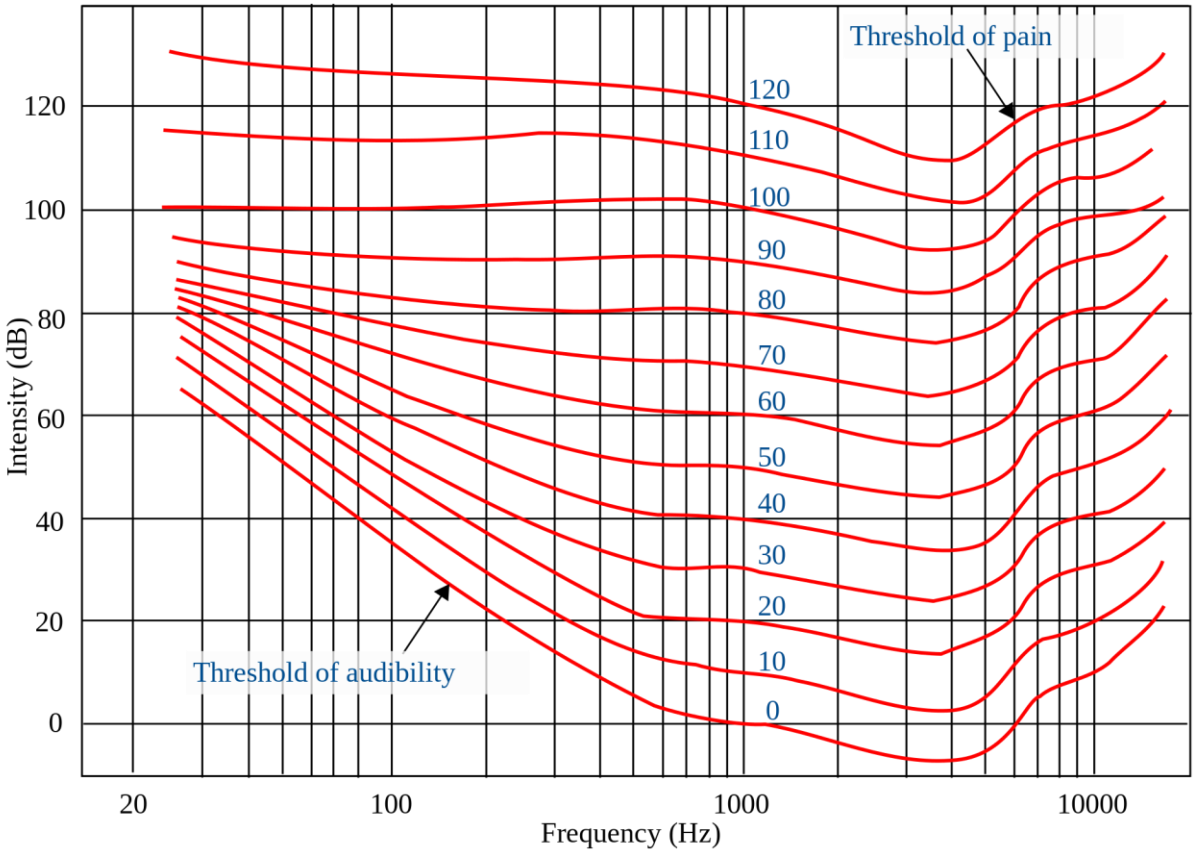
Perception of loudness

- Measured in decibels (dB)
  Logarithmic scale

- The perceived loudness vary as the logarithm of the signal energy
- Perceived twice as loud
  ⇔ level 10 dB greater

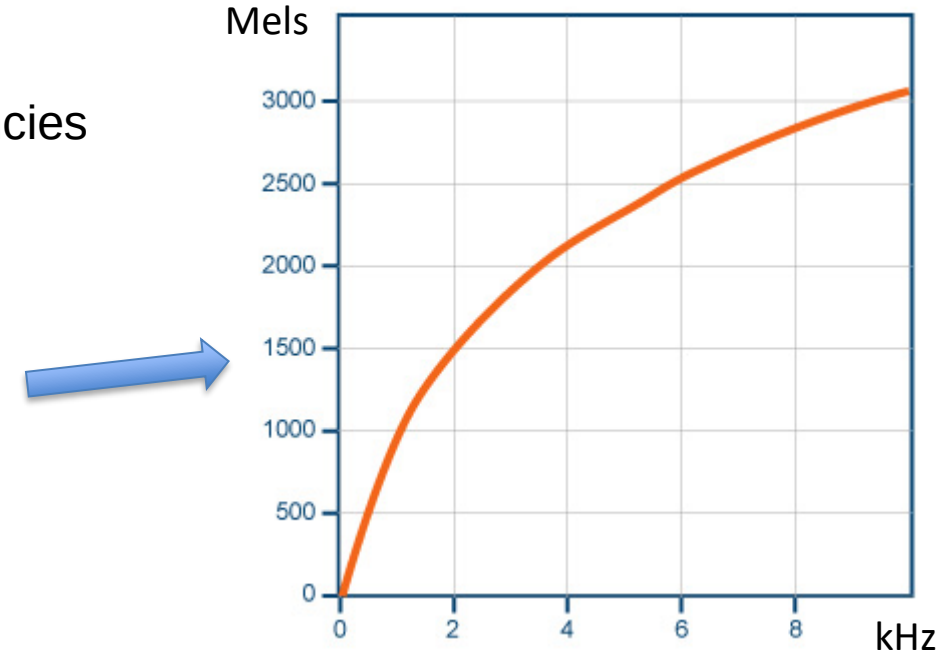


Perception of loudness



Perception of frequency

- The ear is more efficient in the perception of low frequencies than high frequencies
- The « mel » scale represents the perception of frequencies
  - Roughly linear below 500 Hz
  - Logarithmic above



Speech

- Speech production & perception
- **Spectral analysis (time-frequency)**
- Fundamental Frequency & formants
- Sounds of the language
- Speech signal variability

Basics of speech recognition

Automatic speech recognition

Deep neural networks

Extracting other information

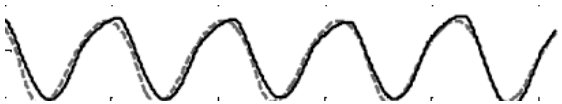
Loria
Laboratoire lorrain de recherche en informatique et ses applications

Spectral analysis (time-frequency)

Speech signal

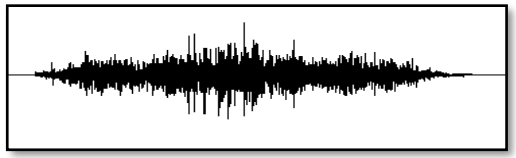
- Speech signal
  - Air flow coming from the lungs and modulated by the vocal tract
  - Gets to the listeners ear as a pressure wave

Source :: $e(t)$



Periodic signal
(vibration of the vocal cords)

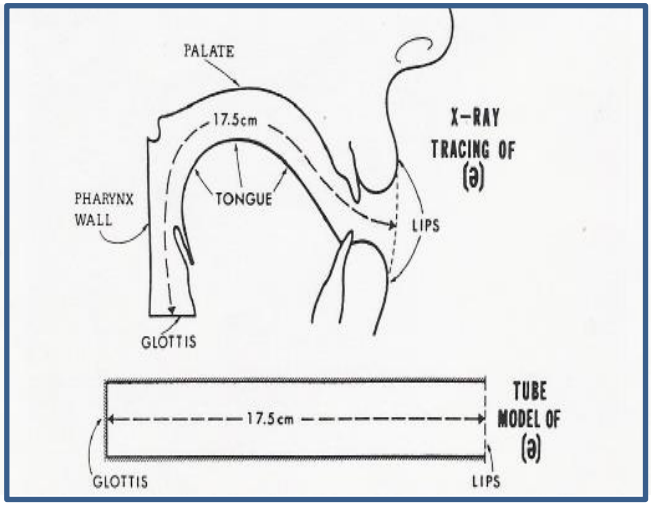
or



Random signal
(when vocal cords **do not** vibrate)

$*$

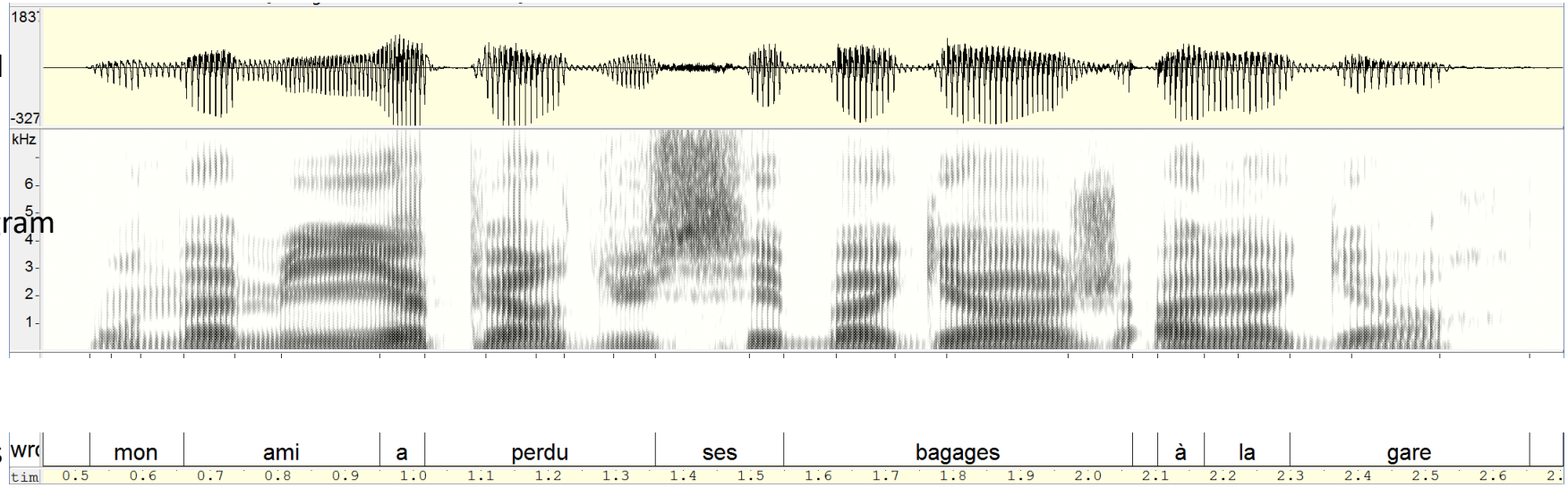
Filter (vocal tract) :: $h(t)$



Speech signal :: $s(t) = h(t) * e(t)$

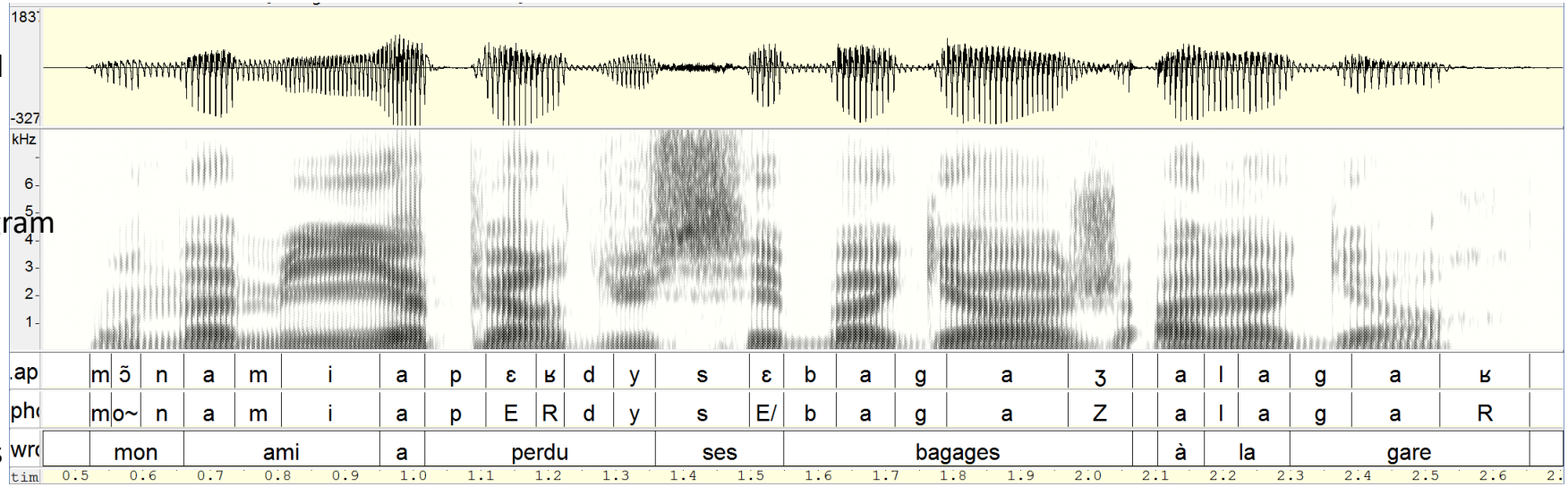
Example of speech signal

- Speech signal ⇔ evolution along time of an electrical signal provided by a microphone
- Spectrogram ⇔ time – frequency representation of the energy



Example of speech signal

- Speech signal ⇔ evolution along time of an electrical signal provided by a microphone
- Spectrogram ⇔ time – frequency representation of the energy



- Coding phonemes
  - IPA ⇔ International Phonetic Alphabet
  - SAMPA ⇔ « *Speech Assessment Methods Phonetic Alphabet* » - easier usage in programs (7-bits ascii codes).

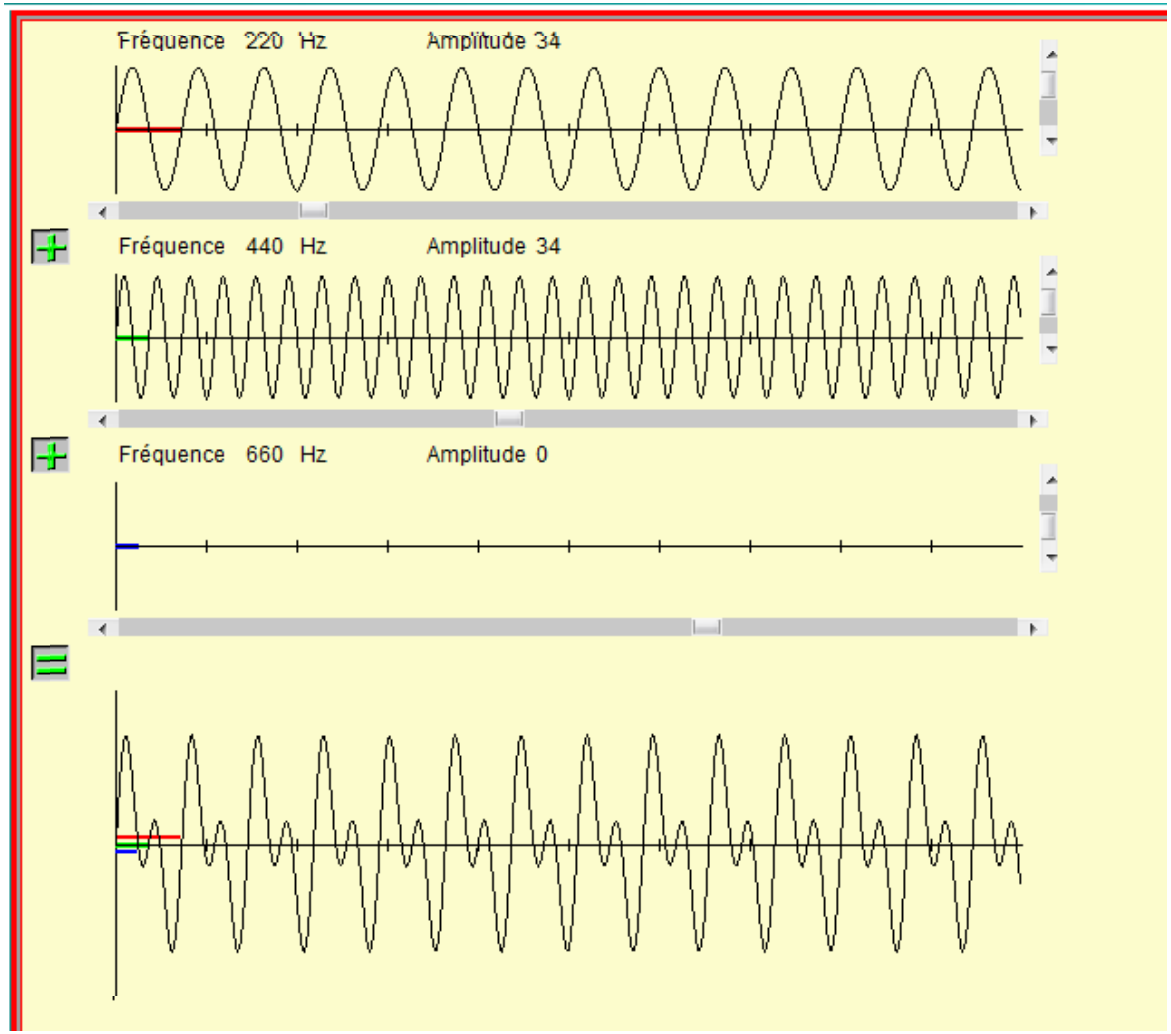
Complex signal

Simple signal

Simple signal

Simple signal

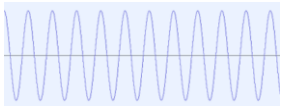
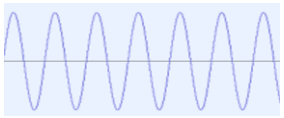
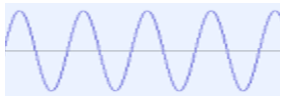
Complex signal



Complex periodic signal = sum of simple signals (sine waves)

Frequency-domain representation

High frequency = high pitch
Low frequency = low pitch

- 5kHz 
- 1kHz 
- 440Hz (la) 

Loria

Fourier transform

- Frequency analysis of a continuous signal

- Fourier transform of a continuous real signal $s(t)$
  ➔ complex spectrum

$$S(f) = \int_{-\infty}^{+\infty} s(t) e^{-i2\pi f t} dt$$

- Inverse Fourier transform

$$s(t) = \int_{-\infty}^{+\infty} S(f) e^{i2\pi f t} df$$

Discrete Fourier transform

- Frequency analysis of a discrete-time signal (i.e., finite sequence of equally-spaced samples)
  ➔ discrete spectral representation

- Fourier transform of a discrete-time real signal $s(n)$
  ➔ complex spectrum

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{-i2\pi \frac{k}{N} n}$$

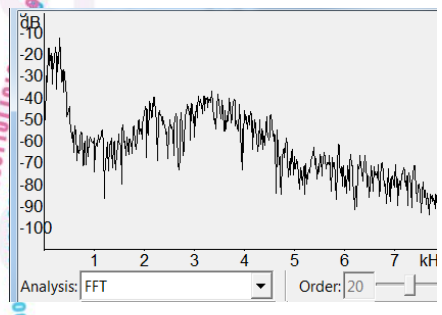
- Inverse transform

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) e^{i2\pi \frac{k}{N} n}$$

- FFT ⇔ *Fast Fourier Transform*
  specific organization of the computations that reduces the amount of operations to be done

Short term Fourier transform

- Consists in splitting the speech signal in « frames »
  and then applying the discrete Fourier transform on each frame
  ➔ time-frequency representation

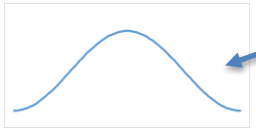


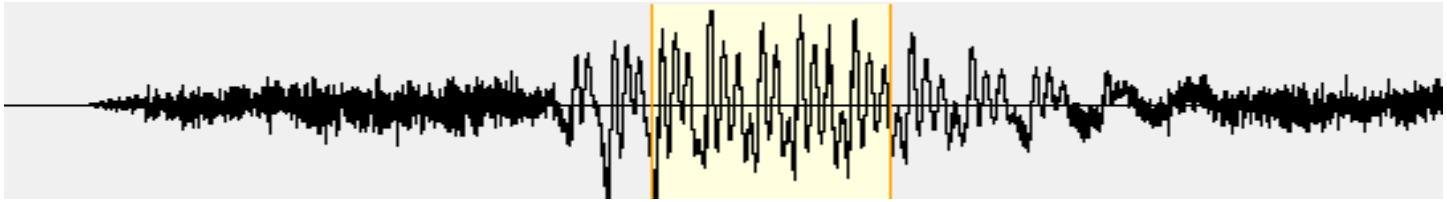
$$S(r,k) = \sum_{n=0}^{N-1} s(rT+n)\,w_a(n)\,e^{-i2\pi\frac{k}{N}n}$$

r : frame index
k : frequency index

Frame

Analysis window
(e.g., Hamming)

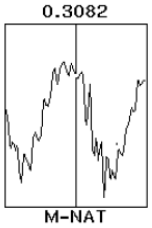


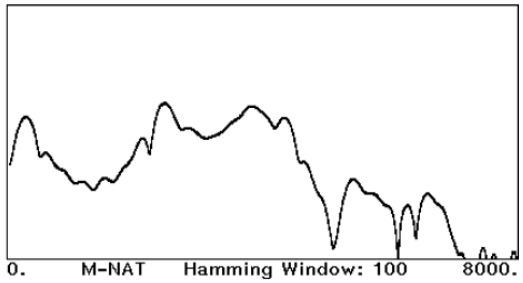


- Example of analysis window ➔ Hamming :: $w_a(n) = 0.54 - 0.46\cos(2\pi\frac{n}{N})$

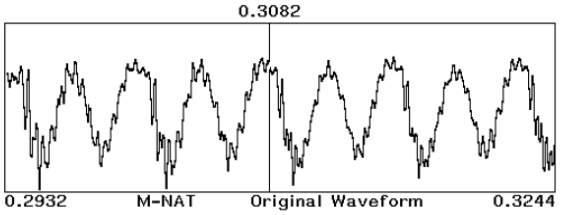
FFT size and frequency precision

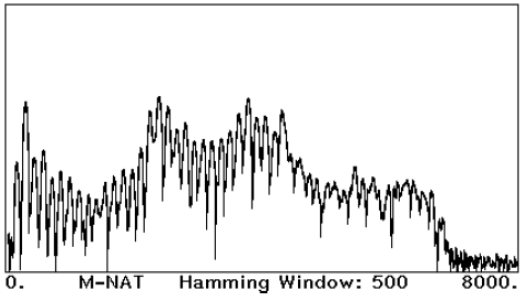
- Narrow window (small number of FFT points)
  - Good temporal precision
  - Bad frequency precision



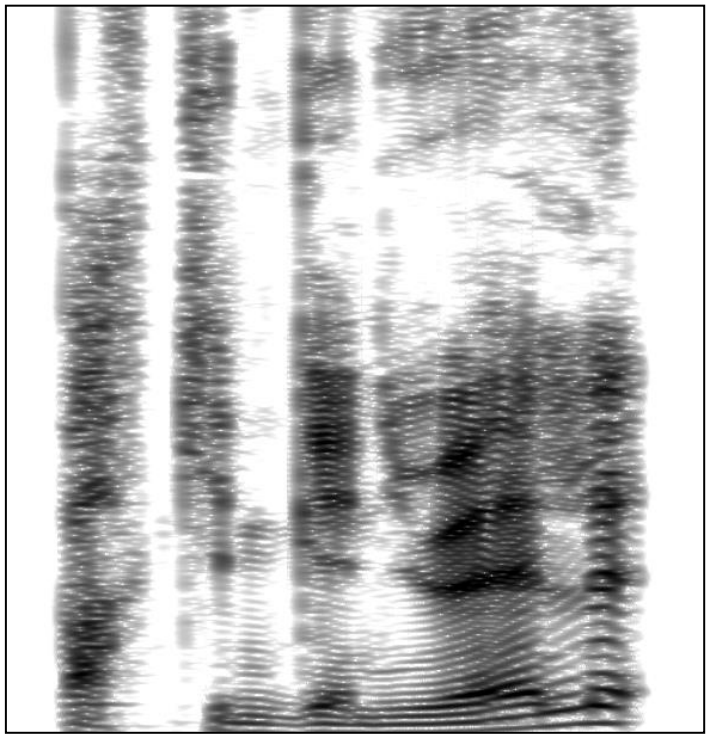


- Wide window (larger number of FFT points)
  - Good frequency precision
  - Bad temporal precision

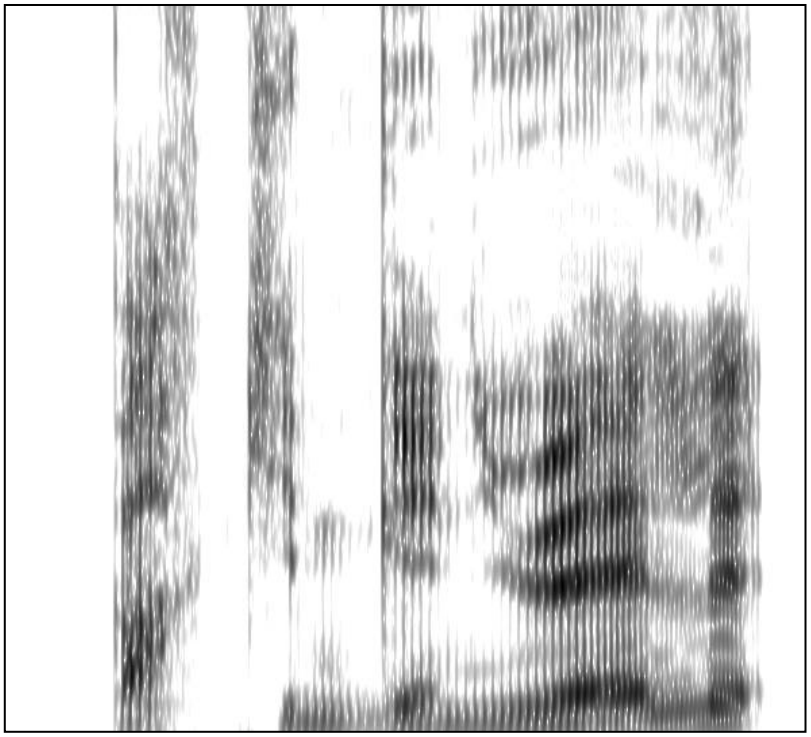




FFT size and frequency precision

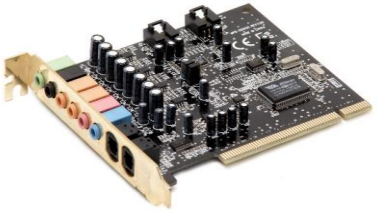


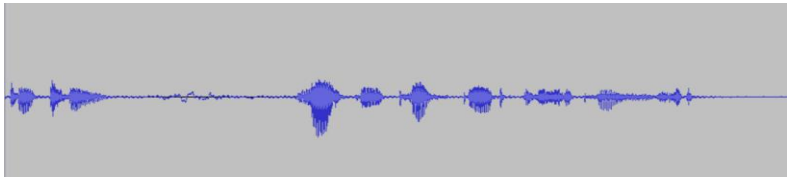
Spectrogram – wide window



Spectrogram – narrow window

Typical recording setup





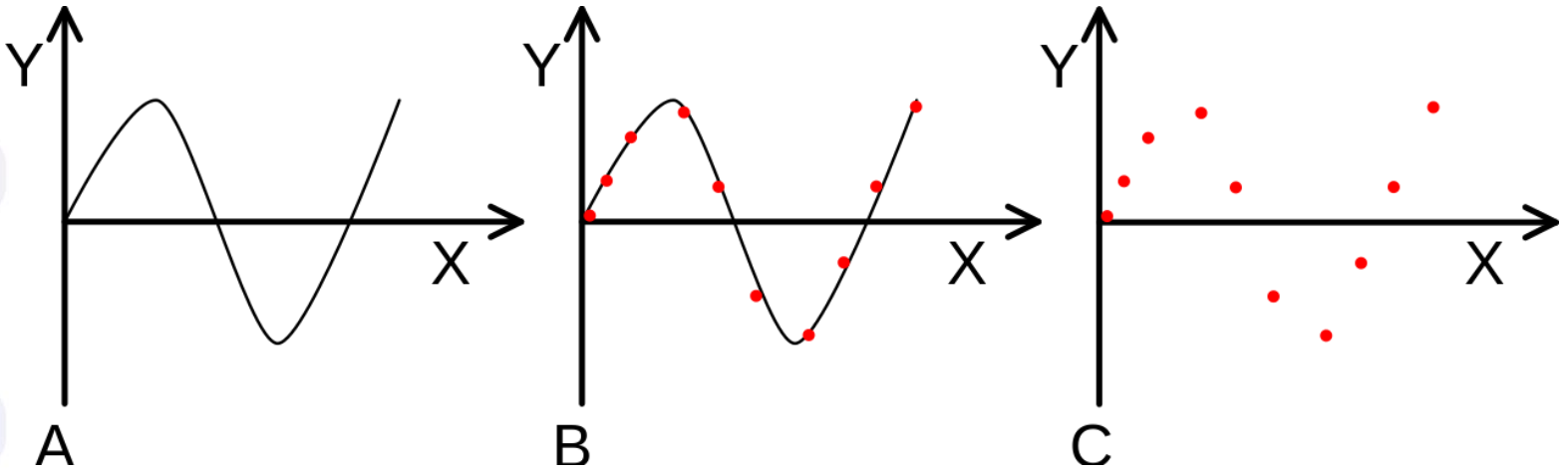
Analog-to-digital conversion

- The analog signal – continuous variable defined for every time value – is converted in a sequence of discrete values

- Analog-to-digital conversion implies
  - **Sampling**
    - Measures at regular time intervals
    - Typically => 8000 Hz, 16000 Hz, 44100 Hz
    - The sampling rate must be greater or equal to two times the highest frequency present in the signal (Nyquist - Shannon)
  - **Quantization**
    - Numerical representation of the values
    - For speech, typically 8 to 16 bits per sample

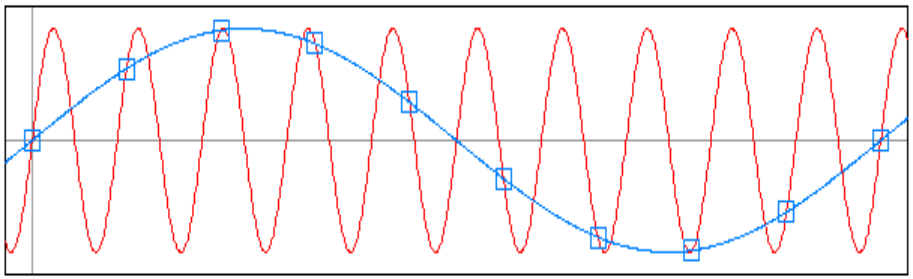
Sampling

- Select samples at regular time intervals
  - Convert the signal to discrete time



Sampling

- The sampling frequency has to be at least twice larger than the frequency bandwidth of the signal (Nyquist Theorem)



Loria

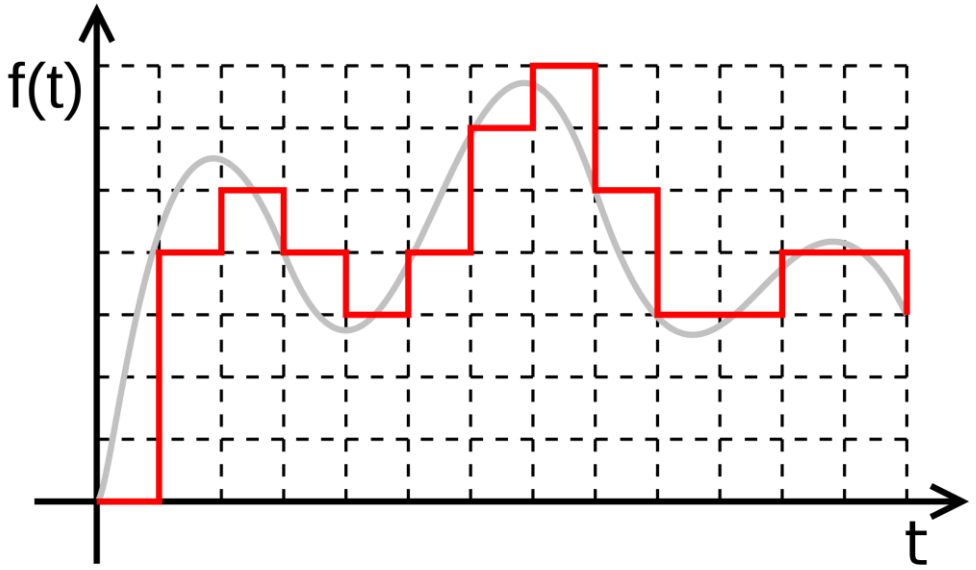
Sampling

- Impact of the sampling frequency on the signal quality:
  - 44100 kHz (CD) 
  - 16kHz (voice) 
  - 8kHz (narrow band telephone) 

Loria

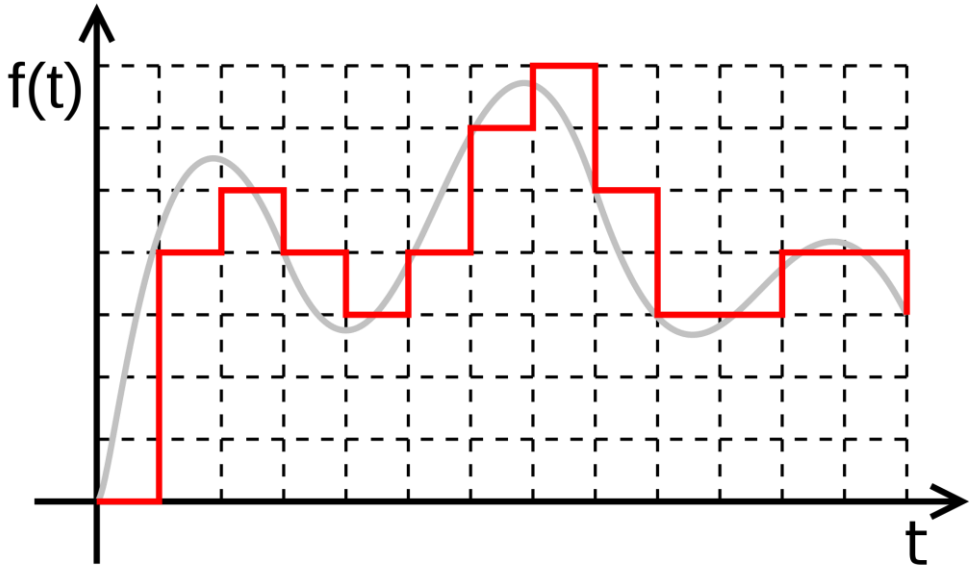
Quantization

- Use a finite number of values for the amplitude



Quantization (binary)

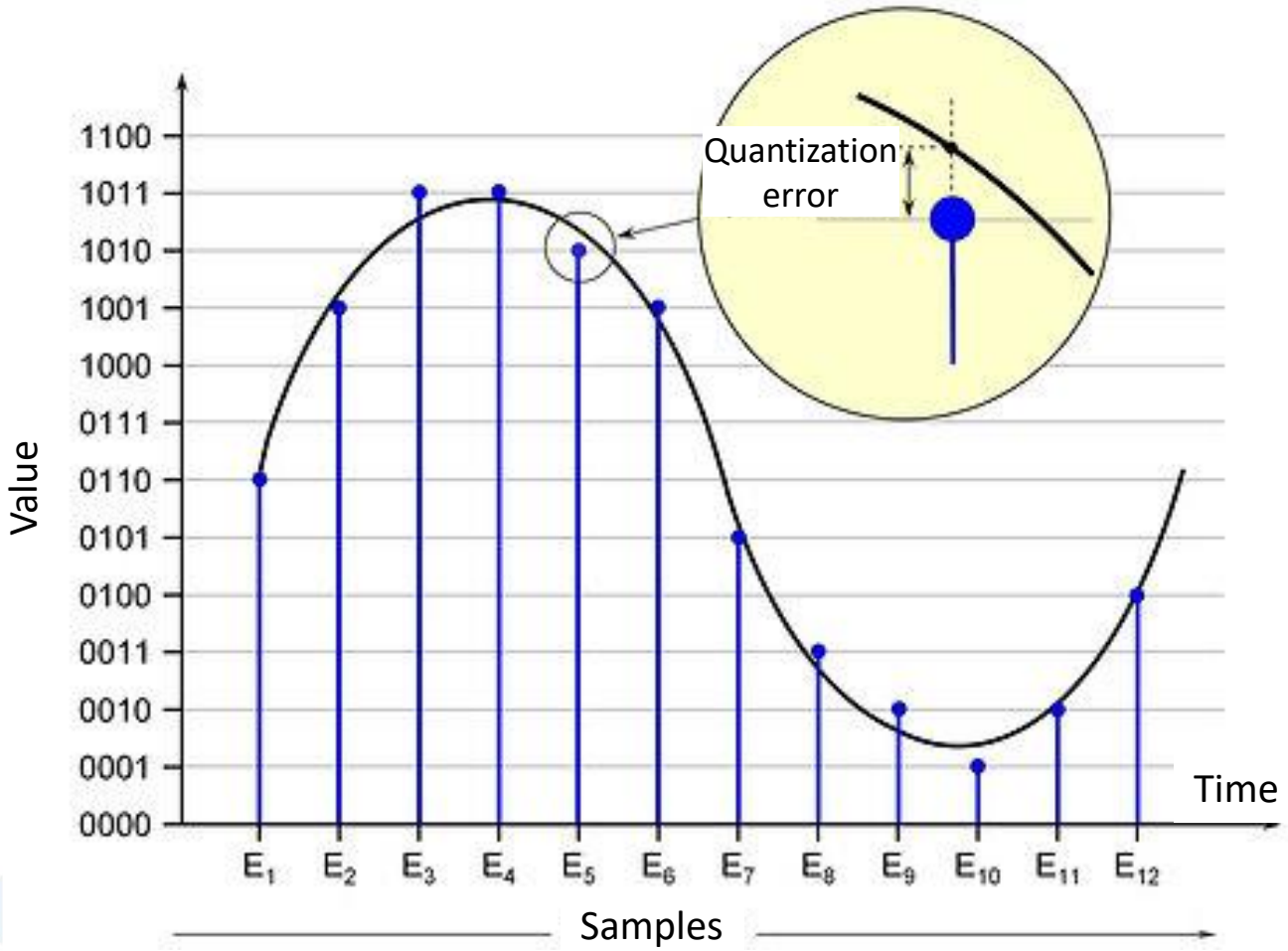
- Binary digit (Bit) :
  - A bit can only take 2 different values: 0 or 1
  - n bits = coding 2^n different values
  - 1 octet = 8 bits
  - Audio : from 16bits to 32 bits



Loria

Quantization error

- The quantization error results from the limited precision (number of bits) available when converting a continuous value to a numerical value



- **Speech**
  - Speech production & perception
  - Spectral analysis (time-frequency)
  - **Fundamental Frequency & formants**
  - Sounds of the language
  - Speech signal variability
- Basics of speech recognition
- Automatic speech recognition
- Deep neural networks
- Extracting other information

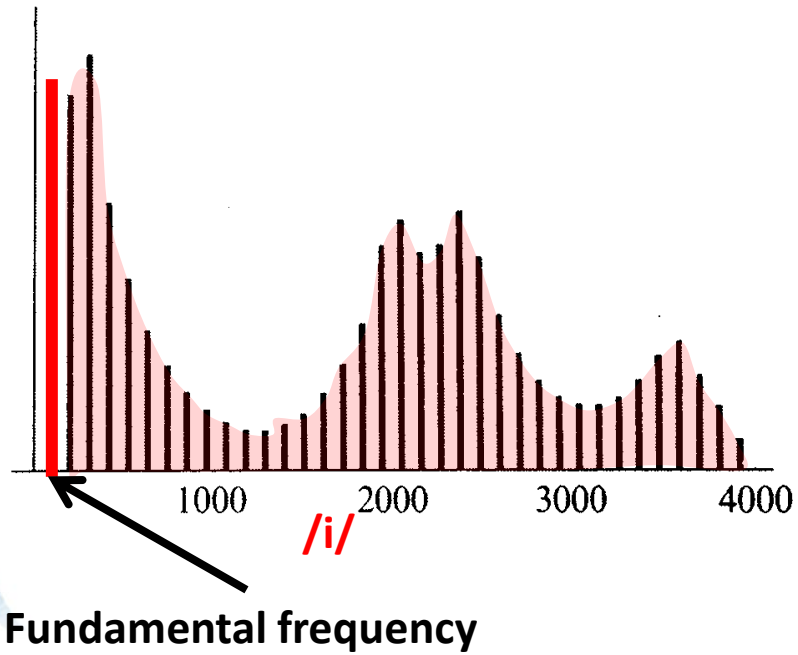
Loria
Laboratoire lorrain de recherche en informatique et ses applications

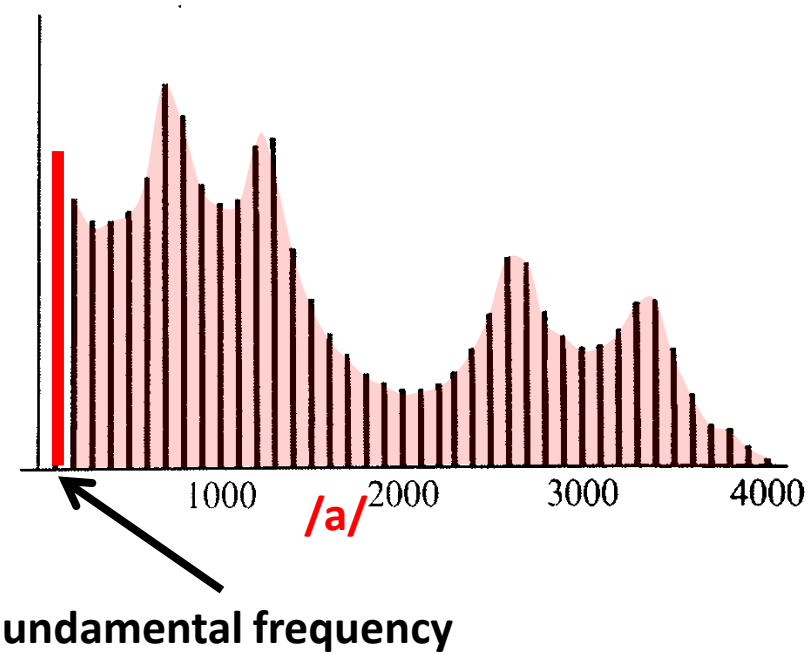
Fundamental Frequency & formants

or, continue with « Basics of speech recognition »

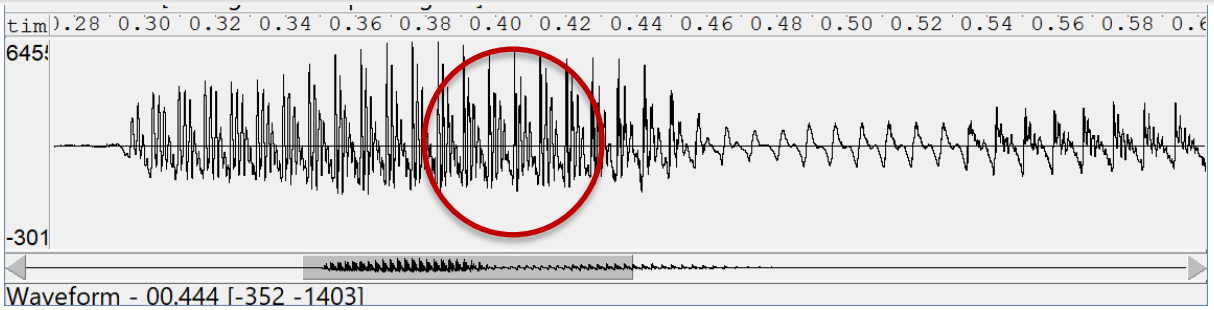
Fundamental frequency

- The fundamental frequency (F0) of a speech signal corresponds to the number of vibrations of the vocal cords over 1 second period
  - Thus, is defined only for « voiced » sounds

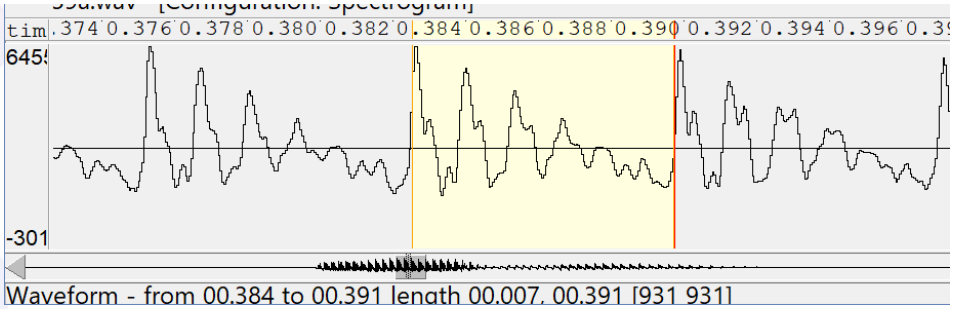




Computing fundamental frequency



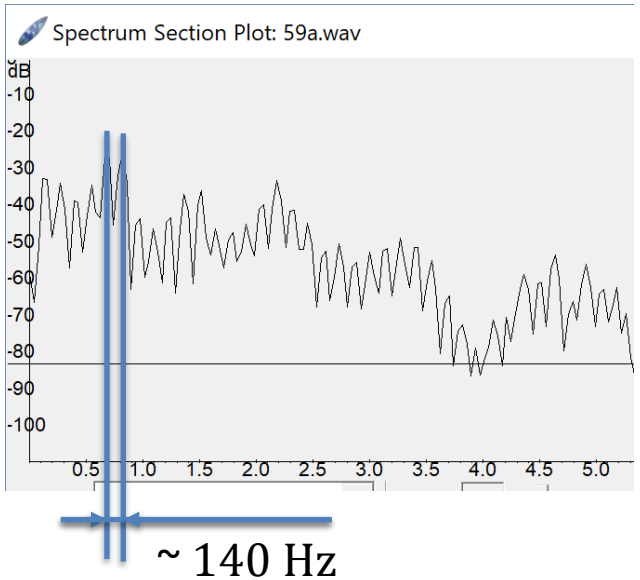
In the temporal domain



➔7ms

➔$\frac{1}{0{,}007} = 144$ Hz

In the spectral domain



~ 140 Hz

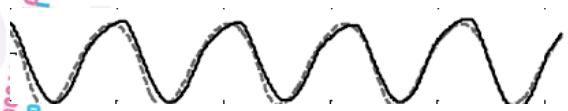
Fundamental frequency & prosody

- Prosodic parameters
  - Fundamental frequency (pitch)
  - Duration of the sounds
  - Energy of the sounds
- Prosody used for
  - Structuring the spoken signal (like punctuation for written texts)
  - Setting emphasis on a word (or group of words)
  - Expressing sentence modality, i.e., question vs. affirmation
- Fundamental frequency also used for
  - Marking the lexical stress (for example, in English)
  - Expressing tone in tonal languages (Chinese, Vietnamese, …)
- Remark: for speech recognition, the fundamental frequency is necessary for processing tonal languages

Formants

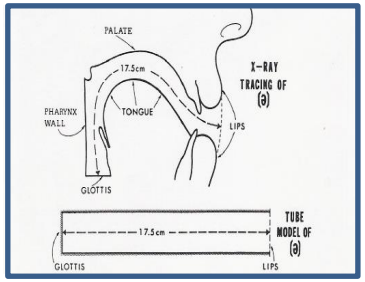
- Correspond to concentrations of energy at specific frequencies that results from acoustic resonances of the vocal tract

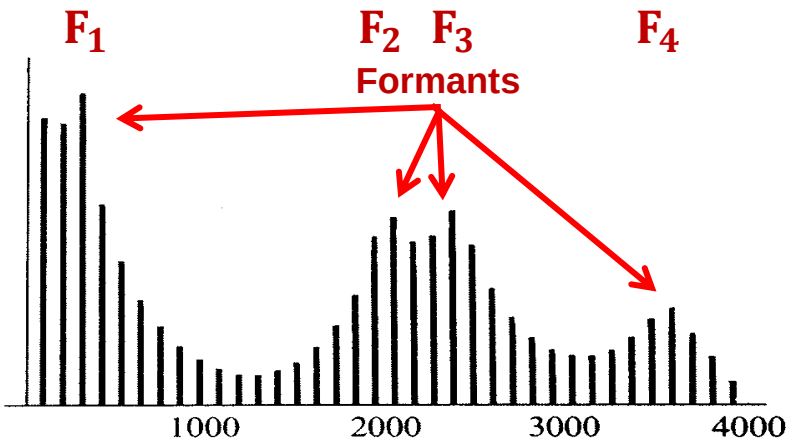
Source



Periodic signal
(when vocal cords vibrate)

Filter (vocal tract)

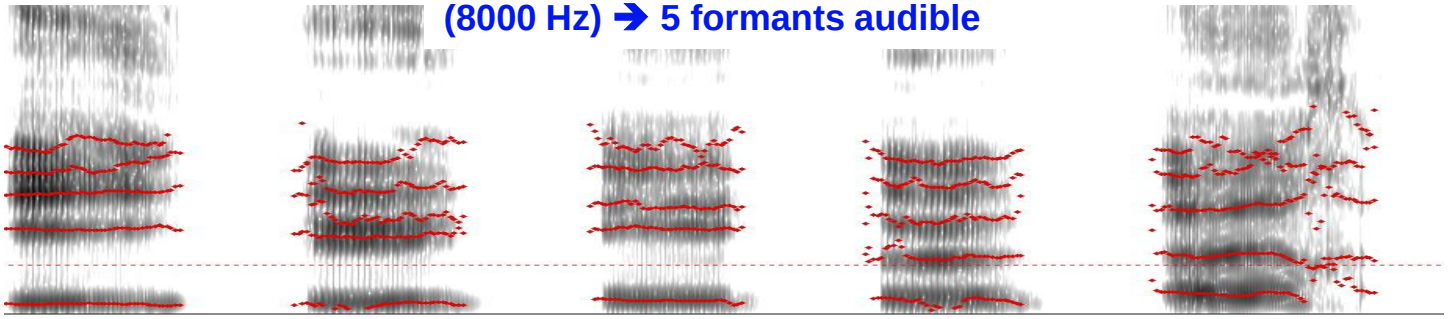




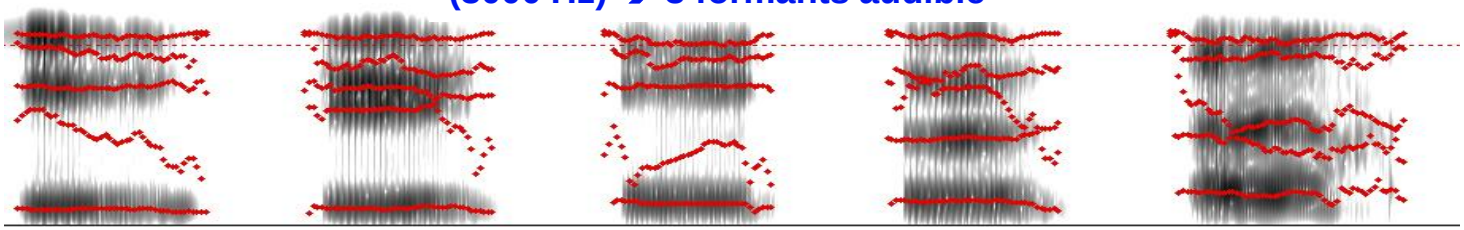
Spectrum, phone */i/*

Impact of bandwidth

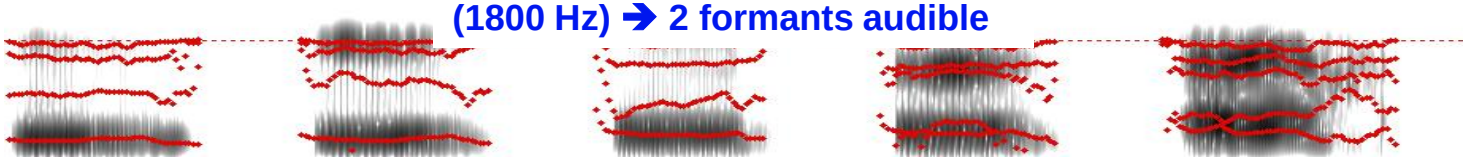
(8000 Hz) ➔ 5 formants audible



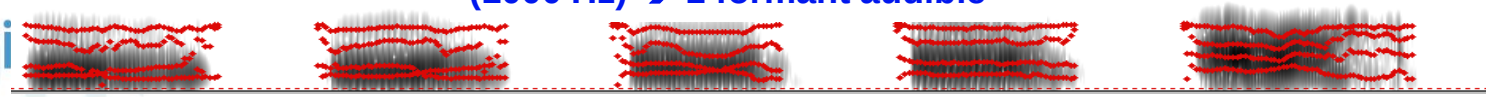
(3000 Hz) ➔ 3 formants audible



(1800 Hz) ➔ 2 formants audible



(1000 Hz) ➔ 1 formant audible



Speech

- Speech production & perception
- Spectral analysis (time-frequency)
- Fundamental Frequency & formants
- Sounds of the language
- **Speech signal variability**

Basics of speech recognition

Automatic speech recognition

Deep neural networks

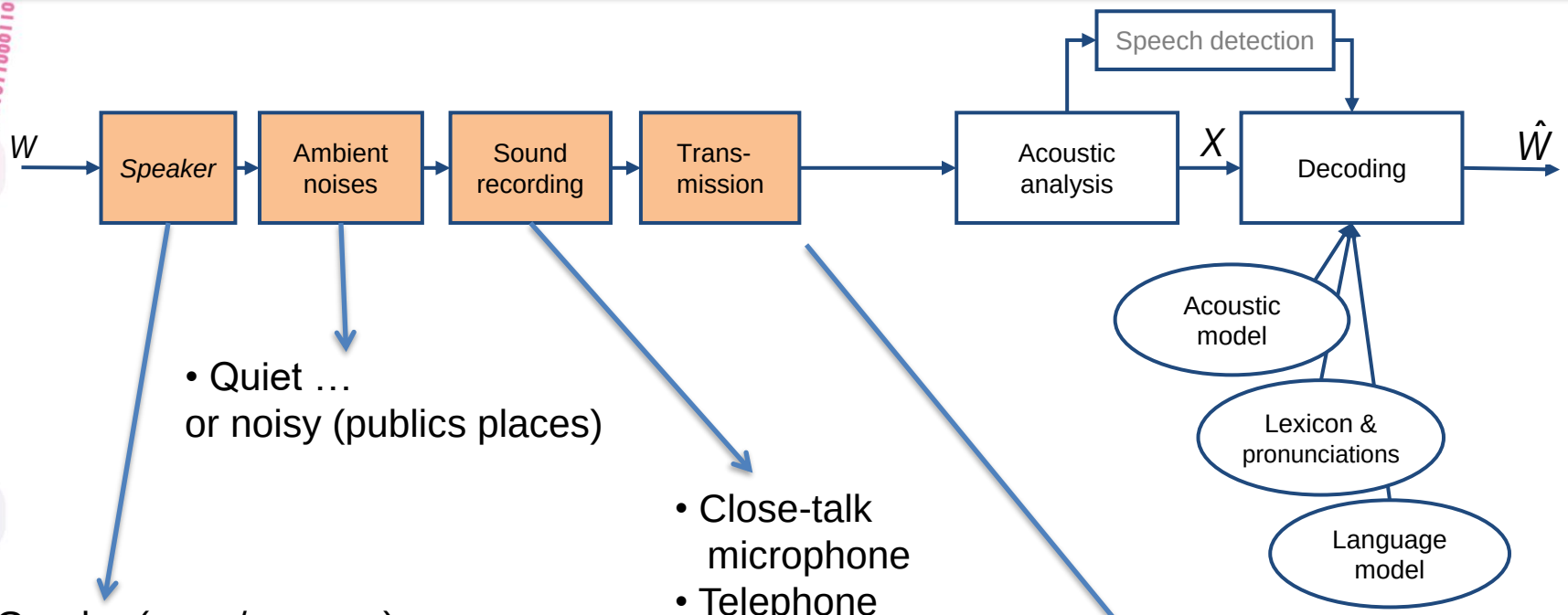
Extracting other information

Loria

Laboratoire lorrain de recherche en informatique et ses applications

Speech signal variability

Variability of the speech signal



- Quiet …
  or noisy (publics places)

- Close-talk microphone
- Telephone
- Hands free

- Gender (man / woman)
- Age (child / adult)
- Accent (none, regional, non-native)
- Voice level (low, …, loud or even shouting)
- Speaking rate (fast, …, slow, over-articulated)
- Emotional state
- Spontaneous speech (disfluencies: hesitations, restart, …)

- Bandwidth
  (telephone vs. wide band)
- Coding
  (analog vs. digital (GSM, VoIP))
- Transmission quality
  (mobile networks, IP)

Pronunciation variants

- A word may have several pronunciation variants
  in particular, in French, with respect to the schwa /ə/ (mute-e), which may be pronounced or not

  verte v e R t
  verte v e R t swa

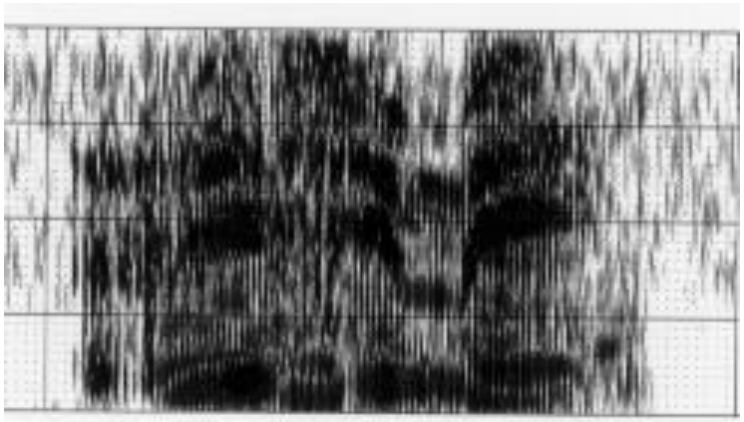
- Also, in French, the word context may lead to the pronunciation of a liaison consonant (when the following word starts by a vowel)

  très t R e
  très t R e z

- In English there are less pronunciation variants, but still variants exist, such as
  - soften ➔ /sɒfn/ /sɔːfn/
  - sorbet ➔ /sɔːbeɪ/ /sɔːrbət/

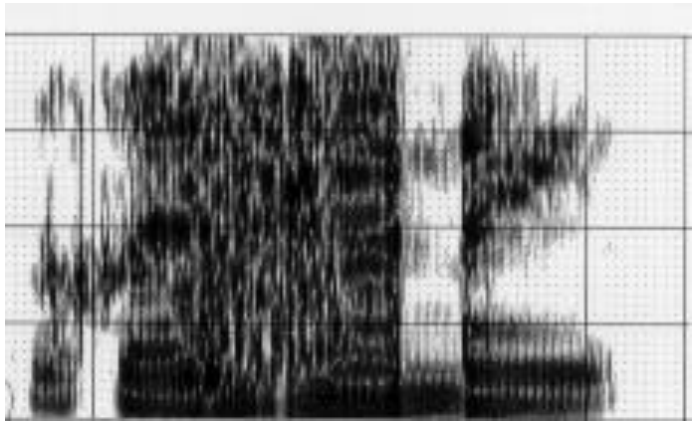
Different voices

Man



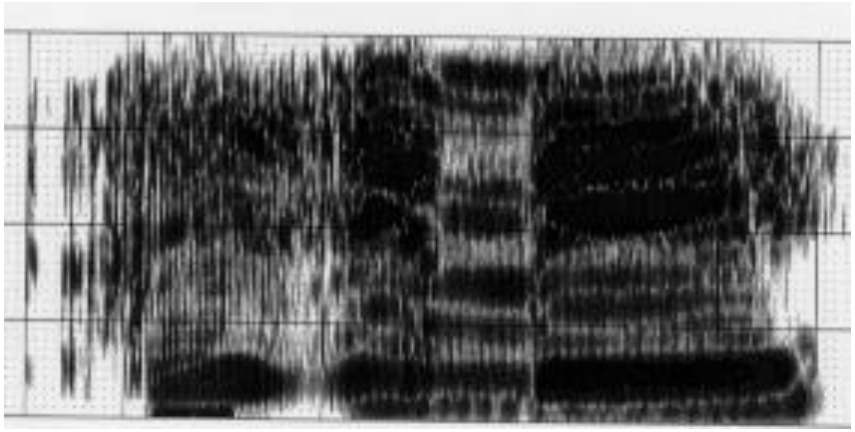
r e z y m e

Woman



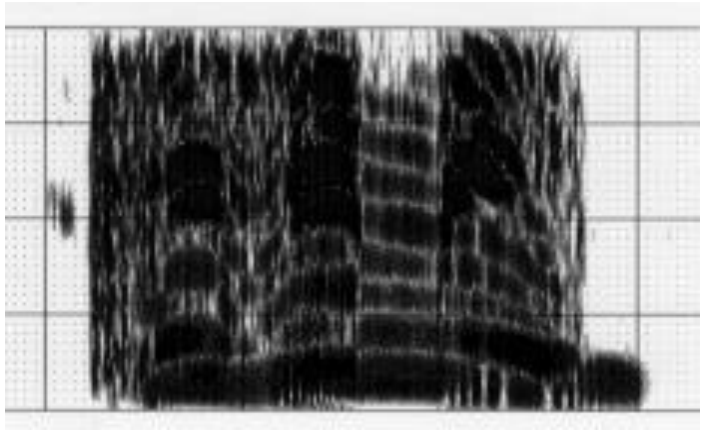
r e z y m e

Teenager



r e z y m e

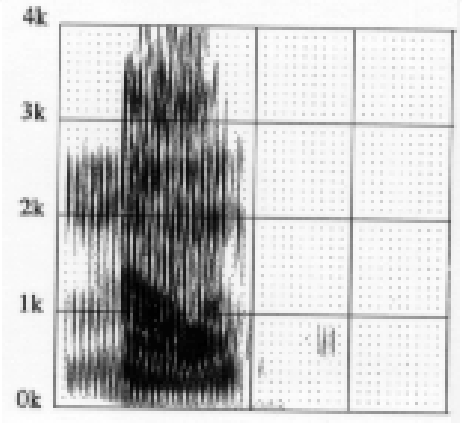
Child



r e z y m e

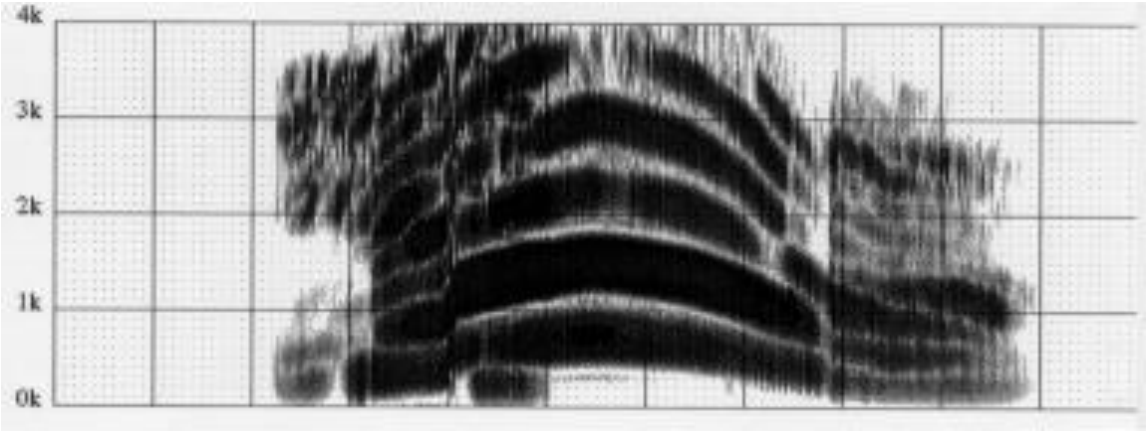
Normal vs. shouting voice

Normal voice



n ɔ̃

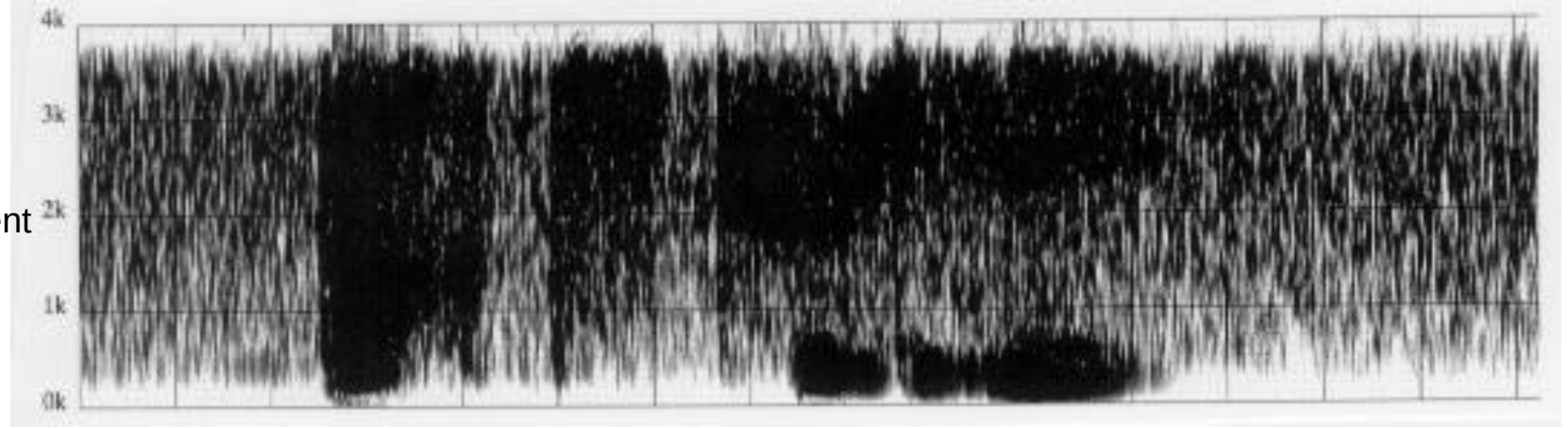
Shouting voice



n ɔ̃

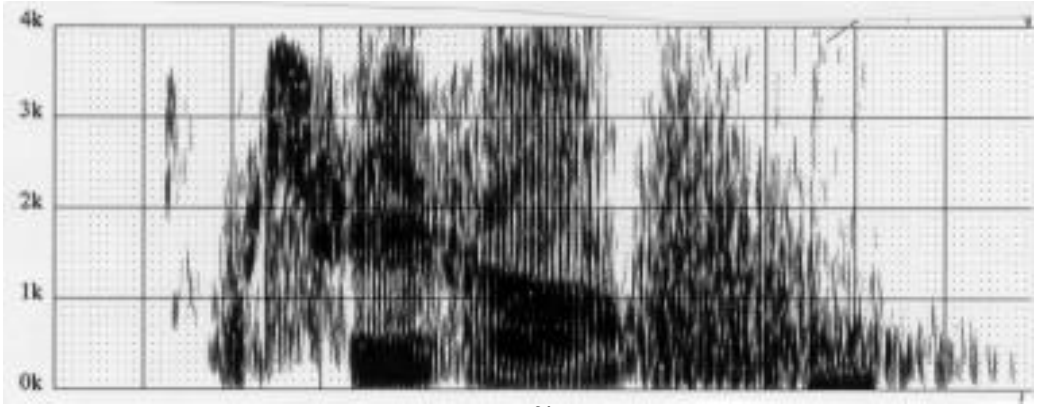
Speech & noise

Environment noise



p a r t i k y l j e

Breathing noise *(before and/or after a mot)*

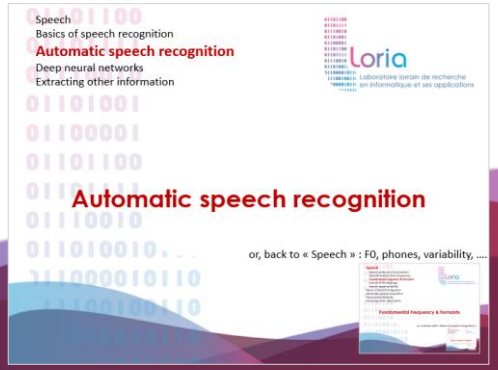


s y i v ã *breath*

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Basics of speech recognition

or, continue with « Automatic speech recognition » : Lexicon, LMs, ….

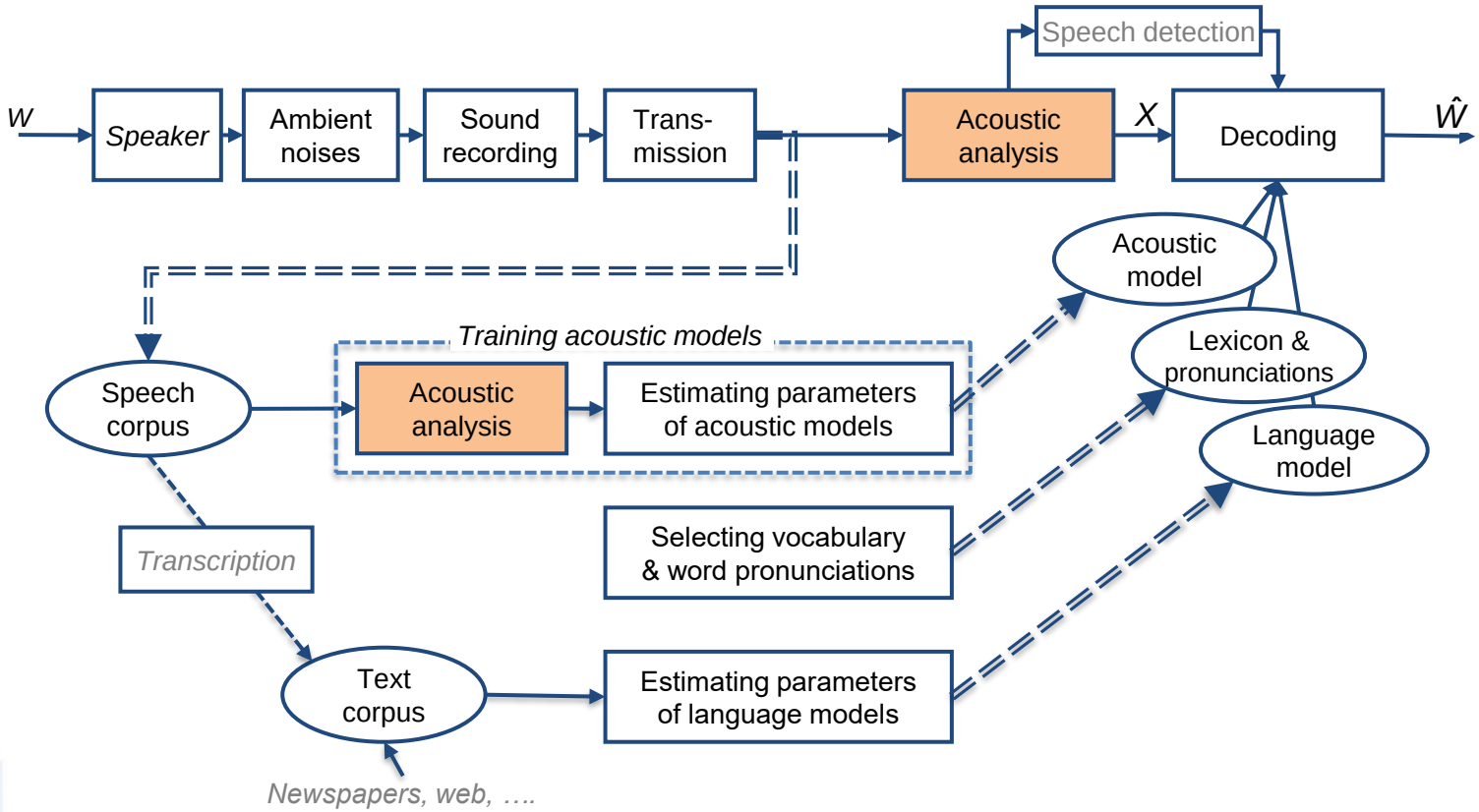


Loria
Laboratoire lorrain de recherche en informatique et ses applications

Acoustic analysis

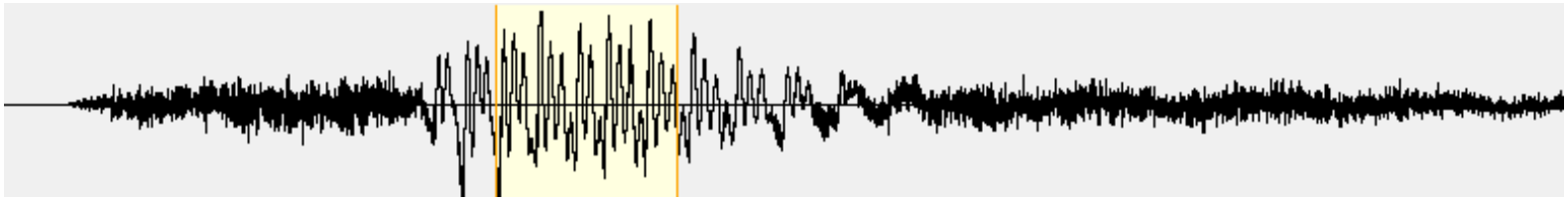
Acoustic analysis

- Measures on the signal, at regular time intervals (typically every 10 ms)



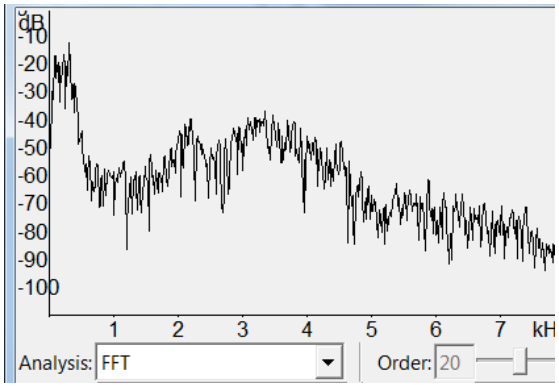
Speech signal … and spectral envelope

- Speech signal

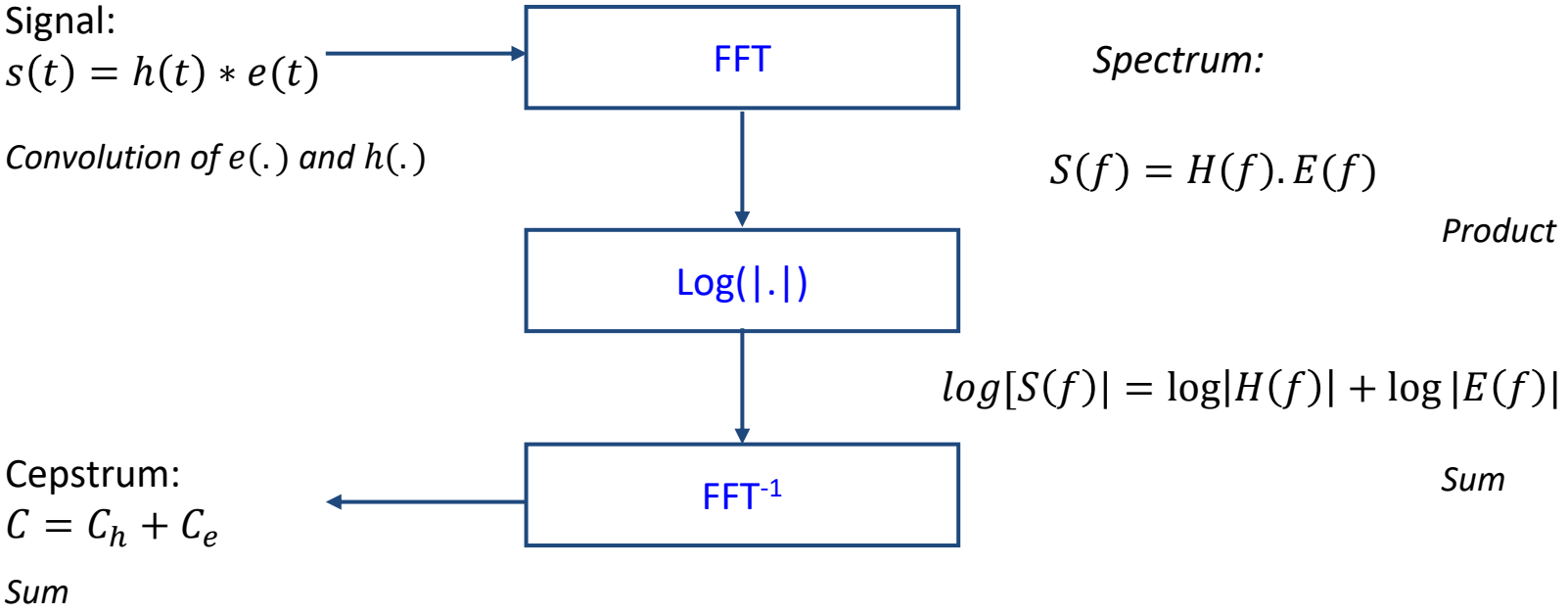


- Spectral envelope of the central slice
  ⇔ energy at different frequencies

FFT spectrum ➔



Cepstrum

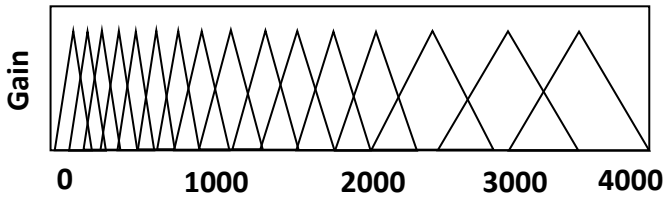


- Lower rank cepstrum coefficients correspond to the vocal tract
- Higher rank cepstrum coefficients can be use to get an estimate of the fundamental frequency

Filter bank

- Signal energy in a set of frequency bands

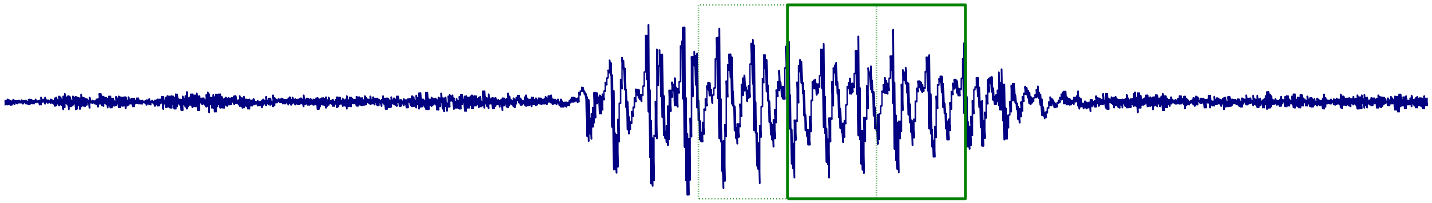
- Characteristics of the filter banks
  ➔ Central frequency of the filters on a
  - Linear scale
  - Perceptive scale (Bark ou Mel)
    - In higher frequencies, filters are larger, and space between central frequencies gets wider
    - Corresponds to the frequency sensitivity of the ear (ability to distinguish small frequency differences in low frequencies)



- Energy in filters can be computed from the STFT (Short Time Fourier Transform) results

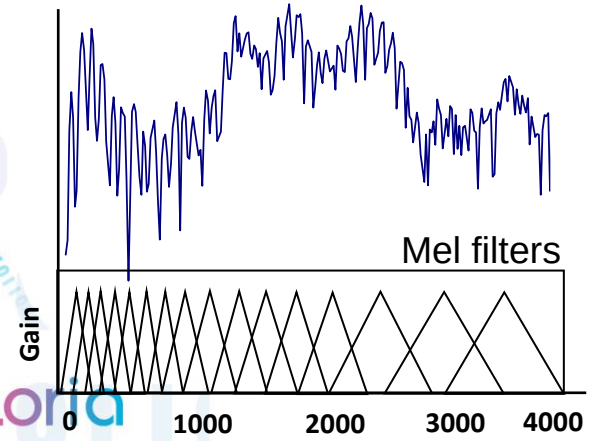
Example of cepstrum analysis

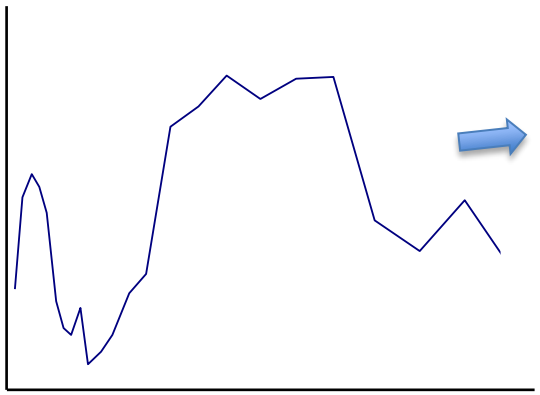
Digital signal (8000 values/second for telephone speech, or 16000, or …)



Processing by blocks (25 to 32 ms time window) *One frame every 10 ms*

→ FFT spectrum ... ... logarithm of the energy for each Mel filter (~24 filters) ... ... cepstral coefficients



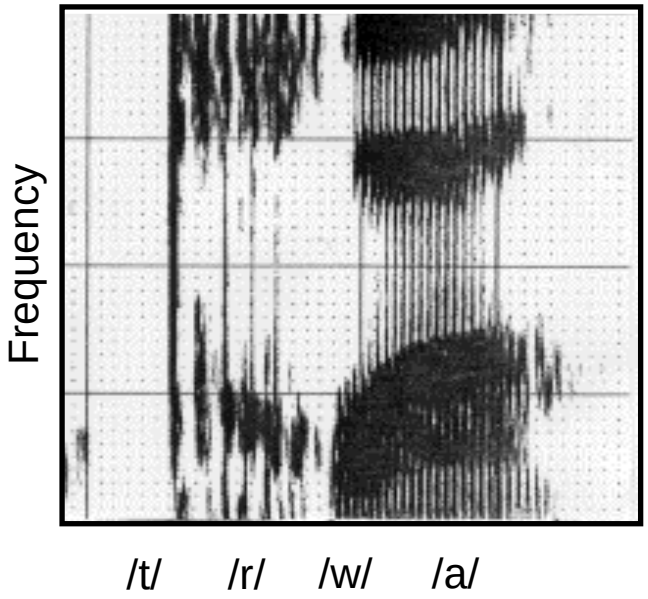


Inverse cosine transform

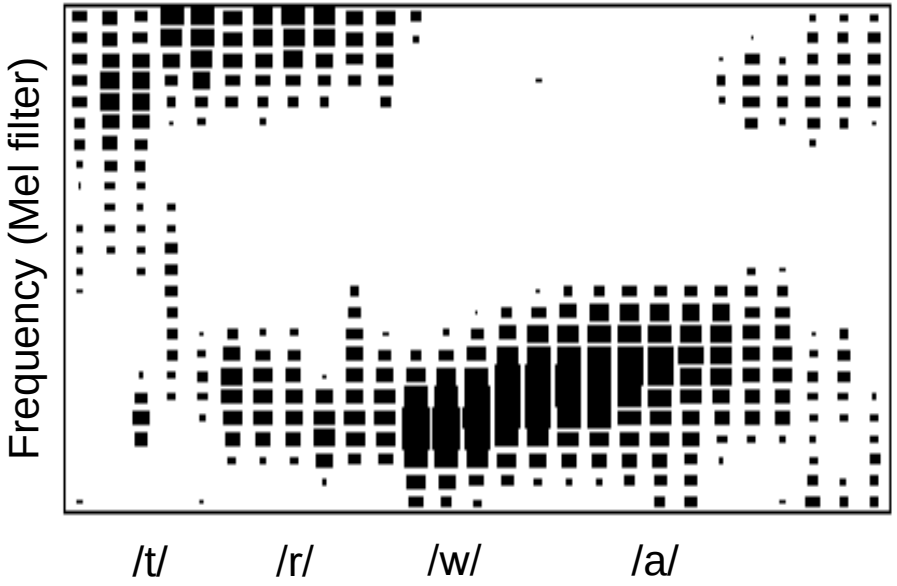
C1, C2, ...
(10 to 13 coeff.)

Spectrogram vs. 10ms Mel spectrum
French word "*trois*" (three)

- Representation of the signal energy with respect to time (horizontal axis) and frequency (vertical axis)



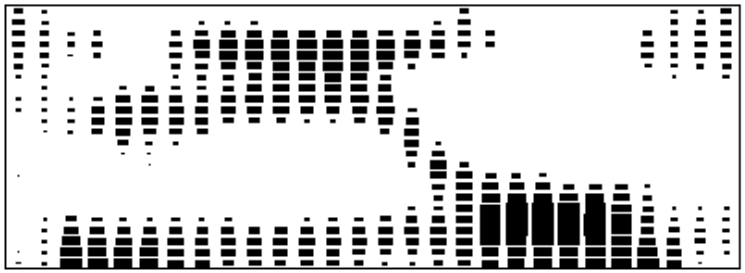
Spectrogram



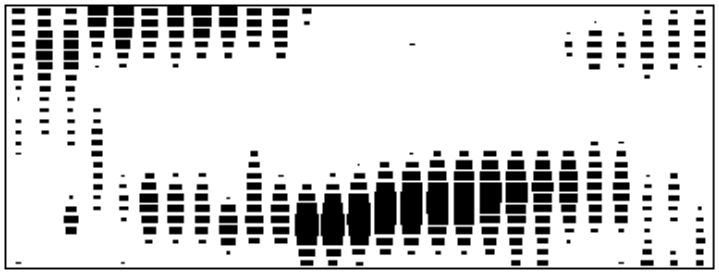
Mel spectrum (frames every 10ms)

Pronunciation of a few French digits

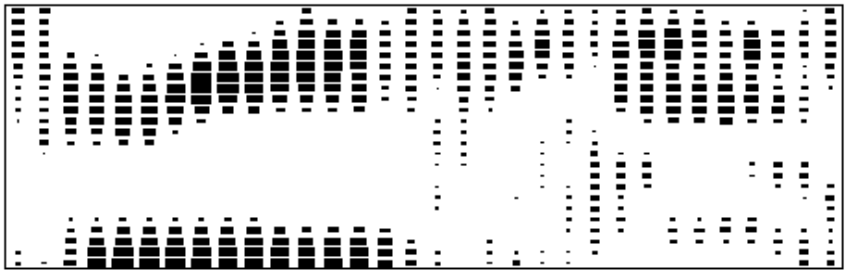
0
"*zéro*"
/zero/



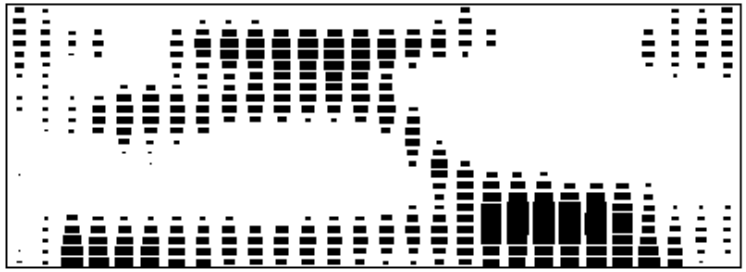
3
"*trois*"
/trwa/

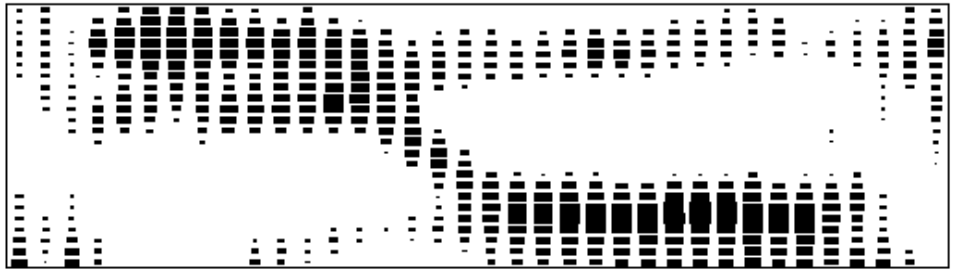


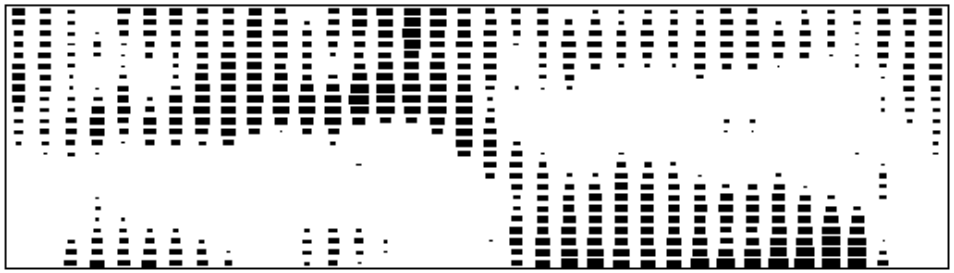
8
"*huit*"
/Yit/



Three pronunciations of French digit "0"







Variability
duration spectrum

Speech

Basics of speech recognition

- Acoustic analysis
- **Acoustic templates**
- Hidden Markov models

Automatic speech recognition

Deep neural networks

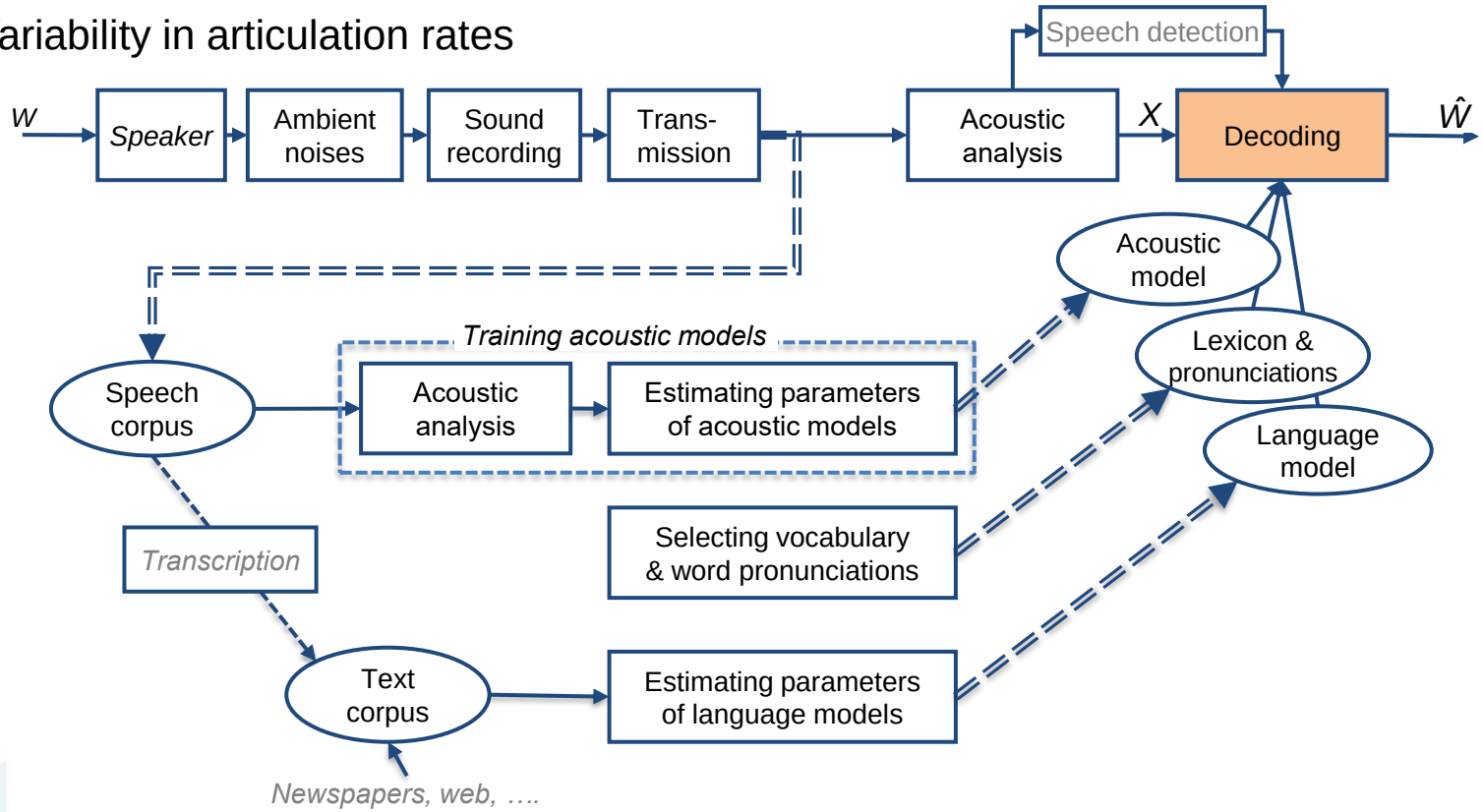
Extracting other information

Laboratoire lorrain de recherche en informatique et ses applications

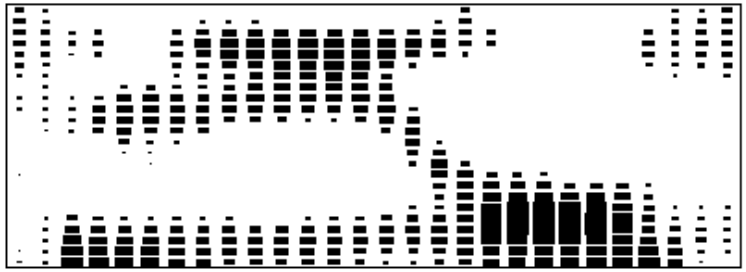
Acoustic templates

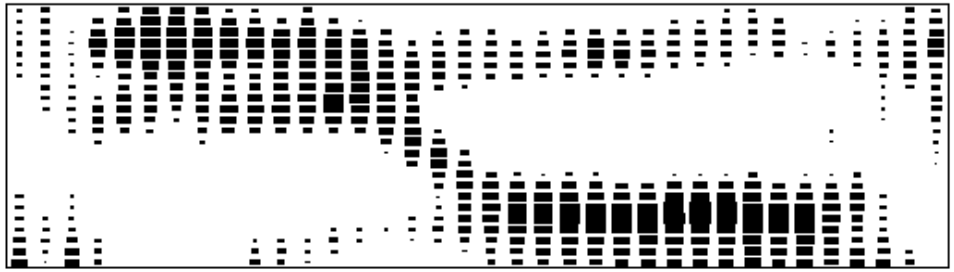
Decoding
based on comparison to acoustic templates

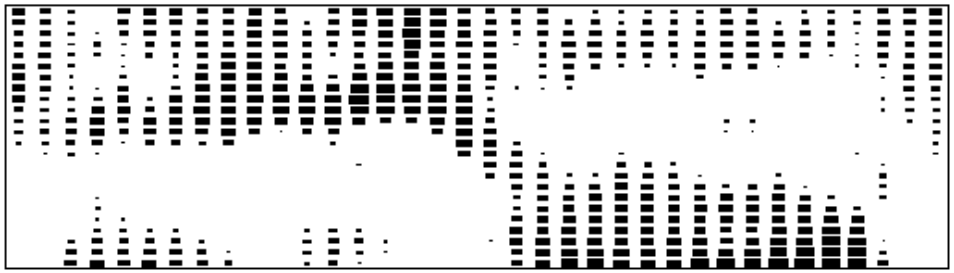
- Compare the acoustic form (sequence of feature vectors) of the word to recognize, to the reference acoustic templates (acoustic forms) associated to the vocabulary words. Need to take into account
  - Differences in durations
  - Variability in articulation rates



Comparing acoustic forms







Variability

duration *spectrum*

Temporal alignment

"distance" between frames

Distortion measures between acoustic frames

- L2 distance between the logarithm of two spectra

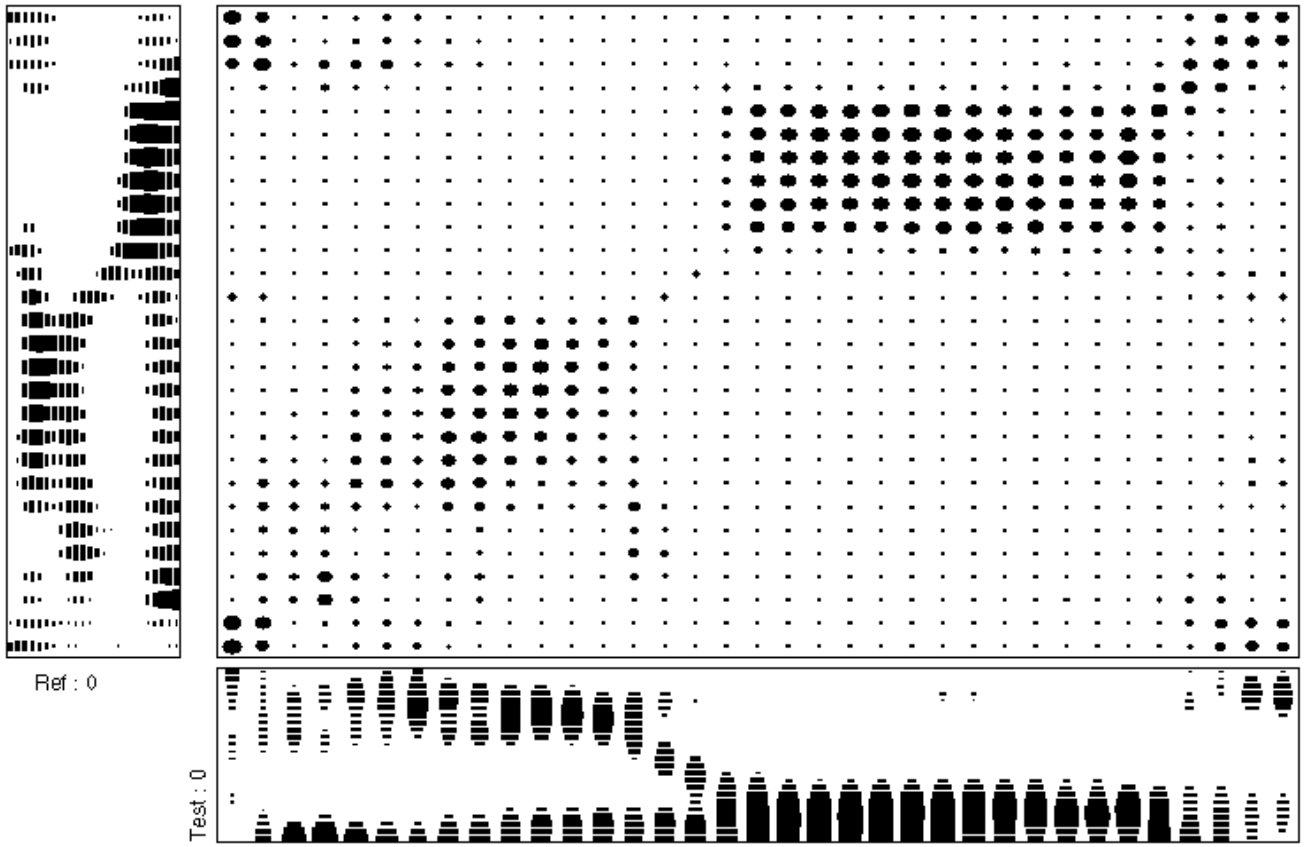
$$d^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\ln f(\omega) - \ln f'(\omega)|^2 d\omega$$

$$= \sum_{n=-\infty}^{\infty} [c(n) - c'(n)]^2$$

- In practice, the cepstral distance is based on a finite number of terms (L)

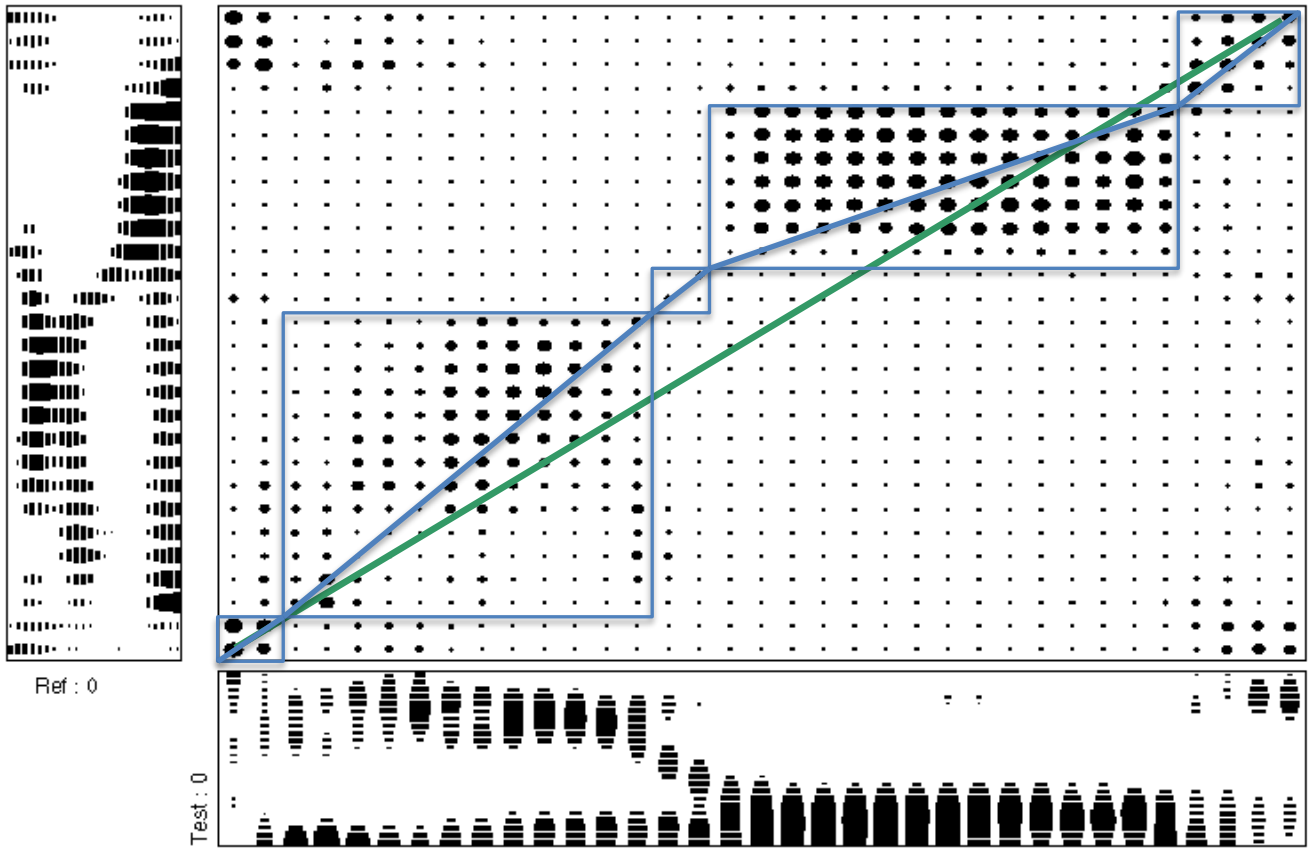
$$d_L^2 = \sum_{n=1}^{L} [c(n) - c'(n)]^2$$

Comparison of two acoustic forms
Local distances (between acoustic frames)



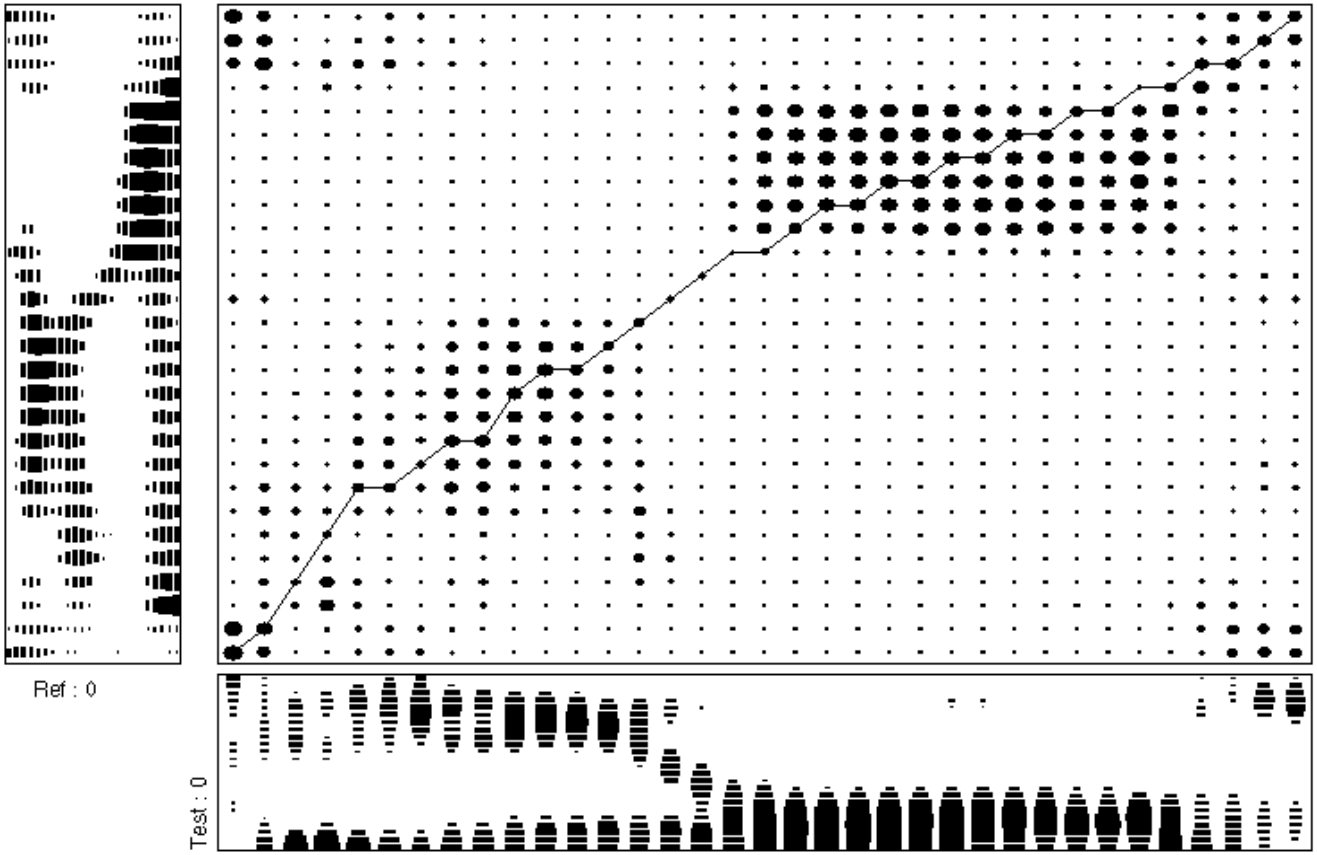
note: the larger the dot is, the more similar are the frames (small distance)

Comparison of two acoustic forms
Problem of non linear temporal distortions



note: the larger the dot is, the more similar are the frames (small distance)

DTW : alignment and global distance



Alignment ⇔ search for the “path" that associate frames that are the most similar

➔ Dynamic programming

(DTW : *Dynamic Time Warping*)

DTW algorithm

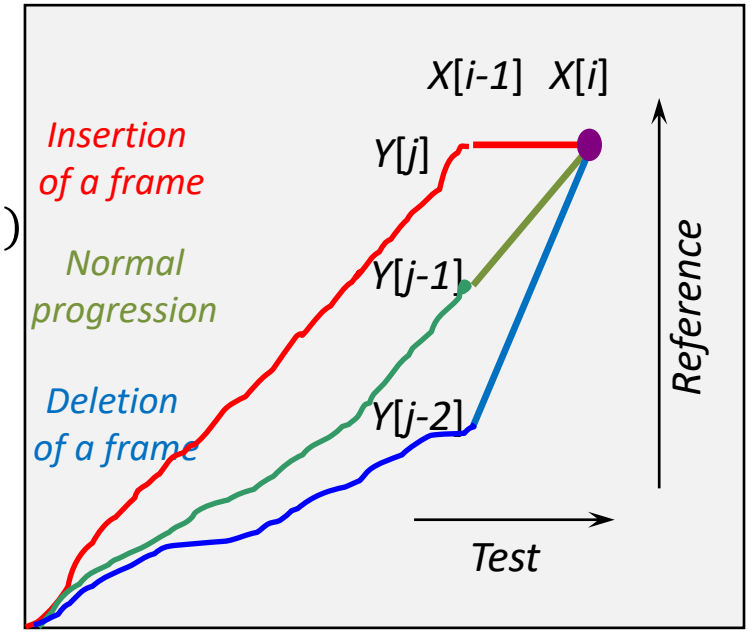
Let define the distance $D(i,j) = \mathrm{Dist}(X[1..i], Y[1..j])$
Between the i first frames of the test $X = \{X[i], i = 1, \dots, I\}$
and the j first frames of the reference $Y = \{Y[j], j = 1, \dots, J\}$

For $i = 1, \dots, I$ (test frames)
  For $j = 1, \dots, J$ (reference frames)

$$D(i,j) = \min \begin{Bmatrix} D(i-1,j) \\ D(i-1,j-1) \\ D(i-1,j-2) \end{Bmatrix} + d(X[i], Y[j])$$

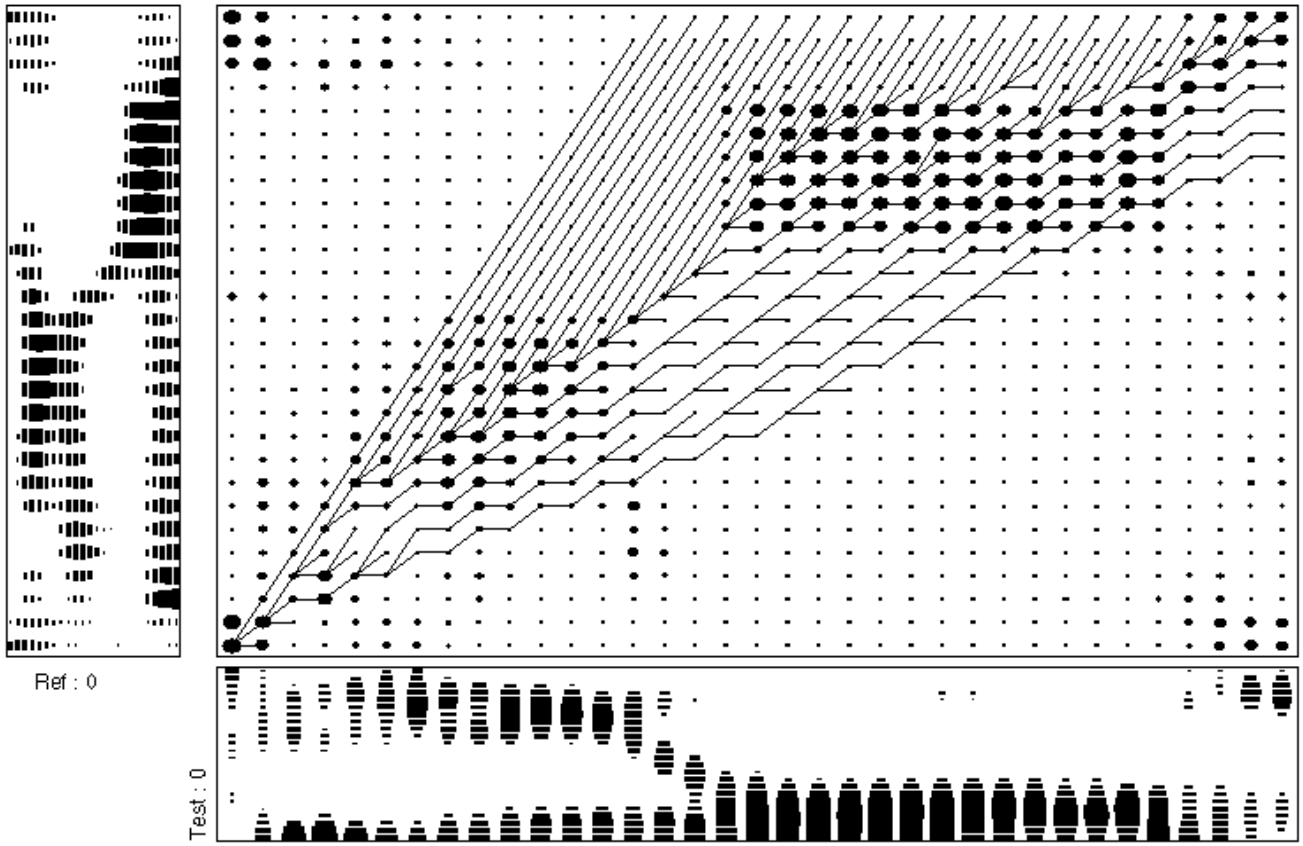
  End for
End for

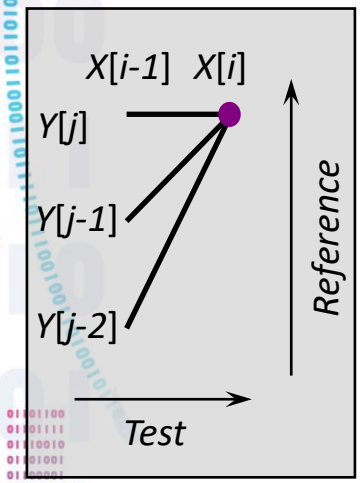
$D(I,J)$ is the global distance



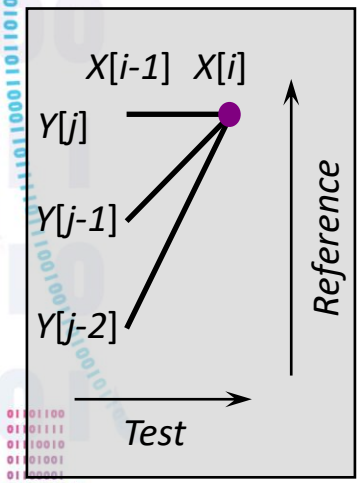
$d(X|i], Y[j])$ is the local distance between frames $X[i]$ et $Y[j]$

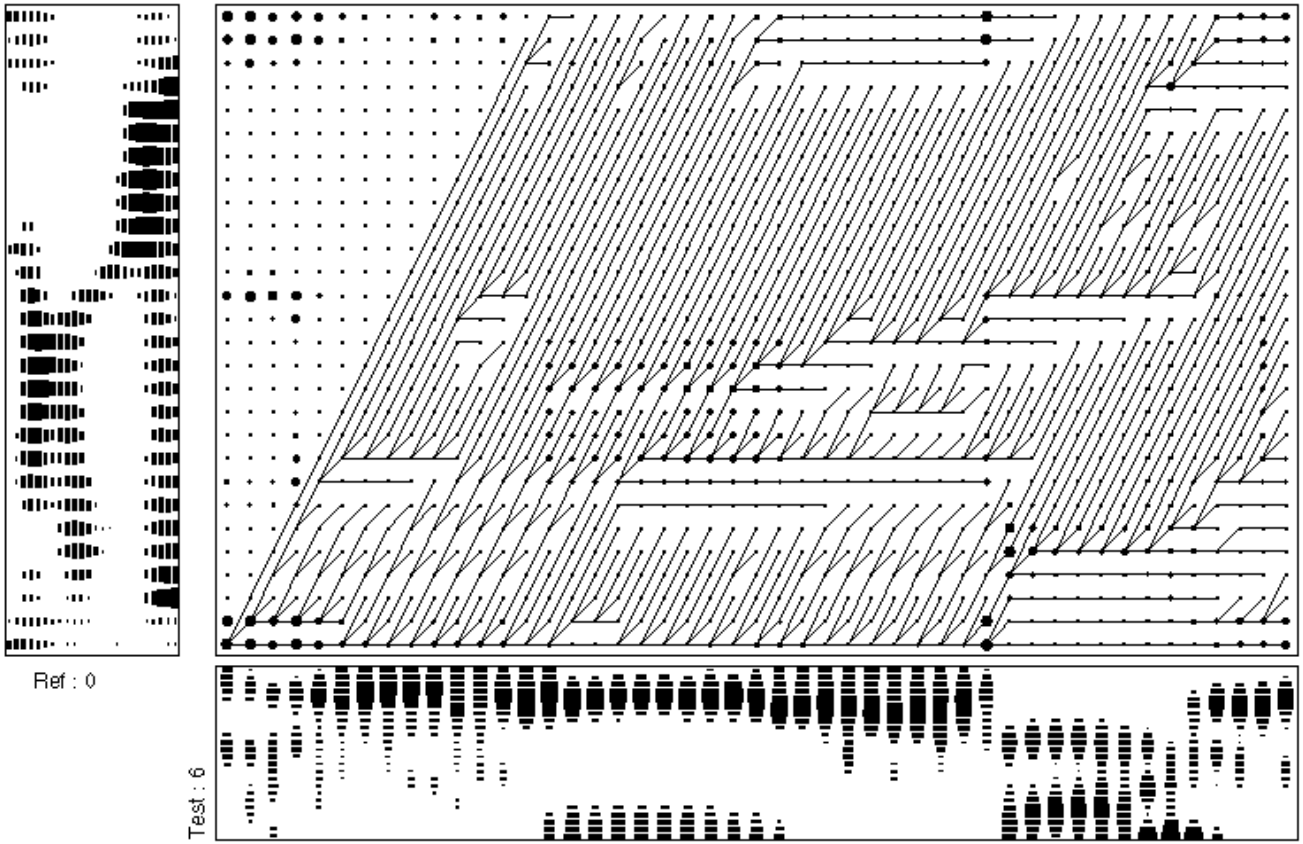
Comparison of “0” with “0”





Comparison of “6” with “0”





Speech

Basics of speech recognition

- Acoustic analysis
- Acoustic templates
- **Hidden Markov models**

Automatic speech recognition

Deep neural networks

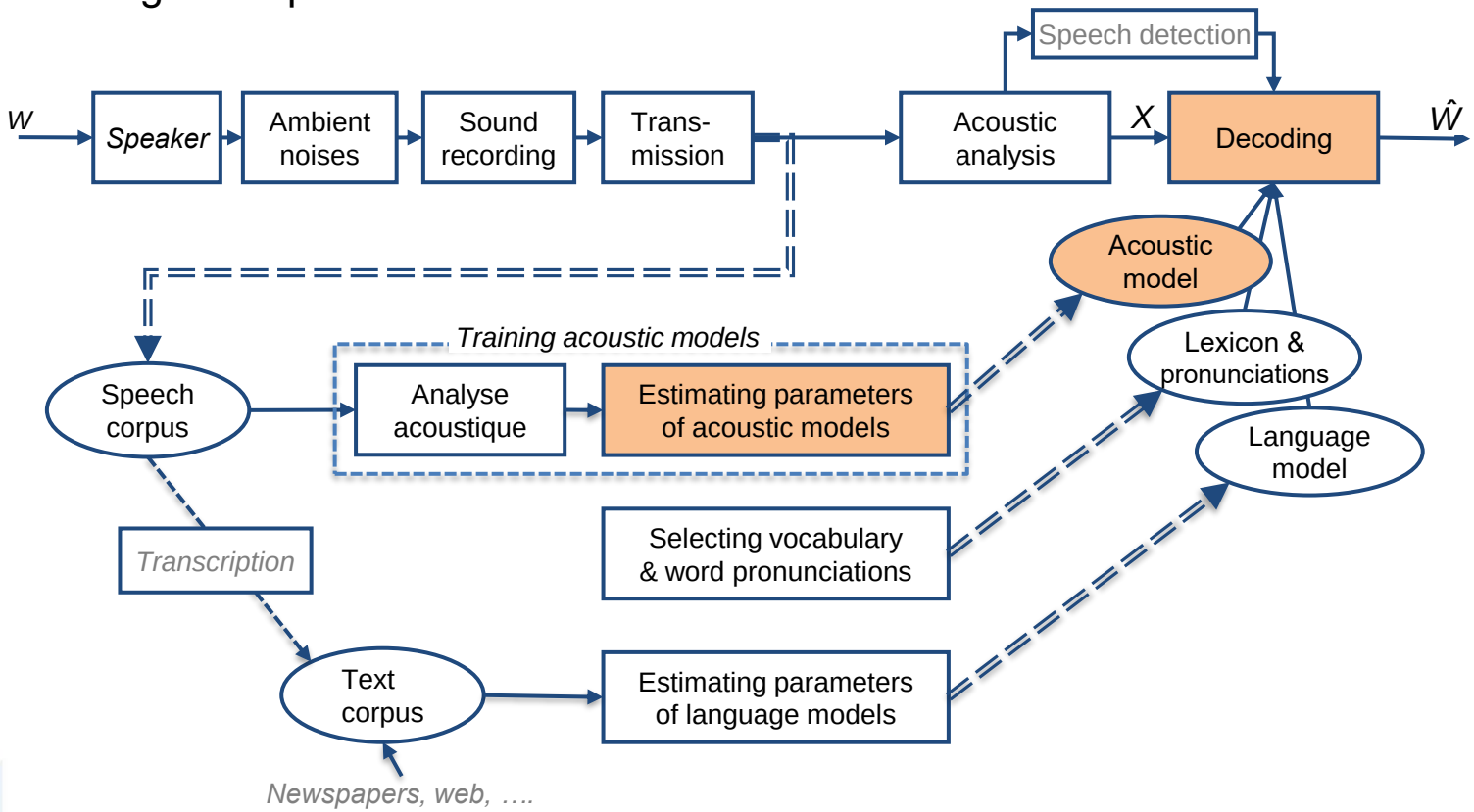
Extracting other information

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Hidden Markov models

Hidden Markov models

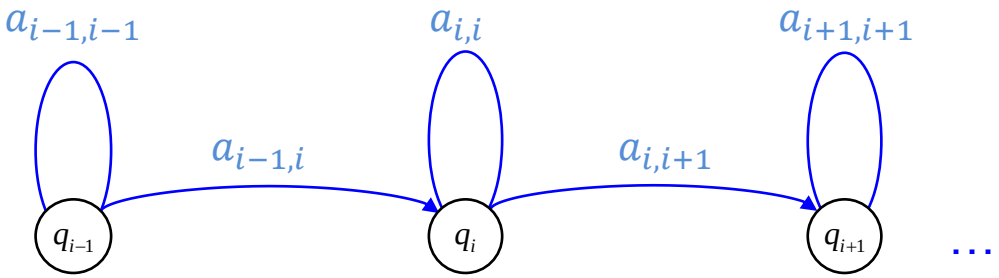
- Statistical representation of the acoustic realizations of a sound or of a word
  - Definition of a hidden Markov model (HMM)
  - Decoding with HMMs
  - Estimating HMM parameters



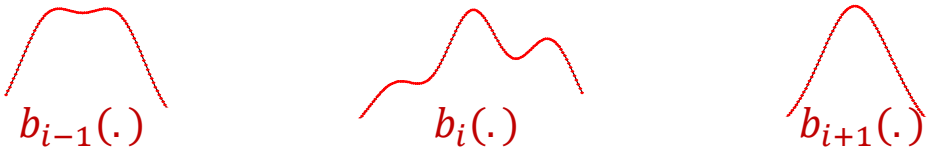
Hidden Markov model (HMM)

Transitions

States



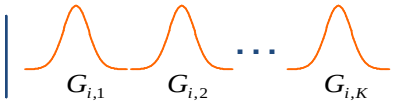
Probability density functions (pdf)



$$P(x_t|q_i) = b_i(x_t) = \sum_{k=1..K} c_{i,k} G_{i,k}(x_t)$$

Observation (acoustic vector)

Gaussian components



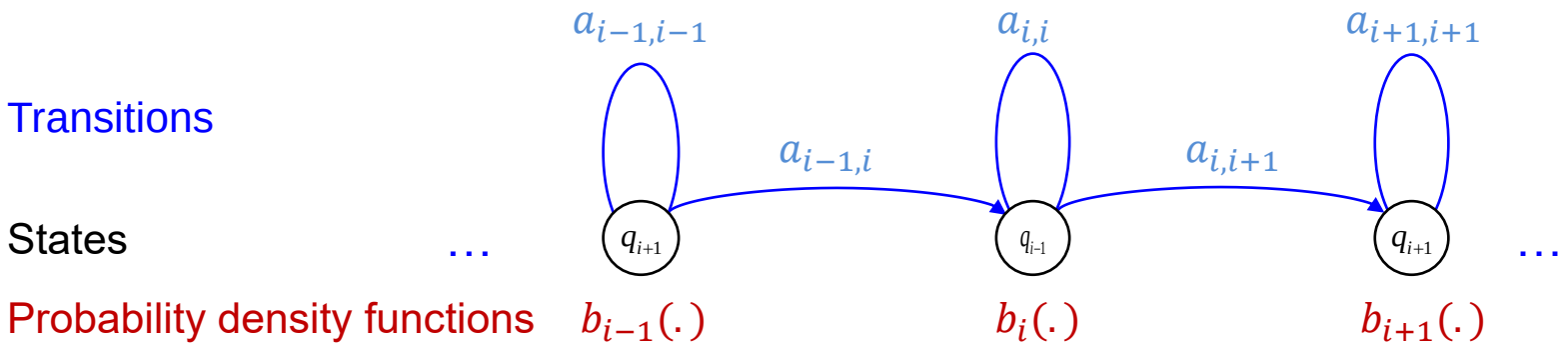
Parameters :

Trans. proba. $a_{i,j}$

Gauss. comp. weight $c_{i,k}$

Gauss. param. $\mu_{i,k}$ & $\Sigma_{i,k}$

Hidden Markov model - parameters



HMMs are defined by two main components

- Underlying Markov chain ⇔ finite automaton + probabilities
  - Initial probabilities $\pi_i = \mathrm{P}(q_1 = i)$, such that $\sum_{i=1}^{N} \pi_i = 1$
  - Transition probabilities $a_{ij} = \mathrm{P}(q_{t+1} = j | q_t = i)$ such that $\sum_{j=1}^{N} a_{ij} = 1$
- Set of probability density functions (pdf) (typically Gaussian mixture models - GMM)
  - pdf $b_i(x_t) = \mathrm{P}(x_t | q_t = i)$
    where $b_i(x_t) = \sum_{k=1}^{K} c_{i,k} G_{i,k}(x_t; \mu_{i,k}, \Sigma_{i,k})$
    $= \sum_{k=1}^{K} c_{i,k} \frac{1}{(2\pi)^{D/2} |\Sigma_{i,k}|^{1/2}} \exp(-\frac{1}{2} (x_t - \mu_{i,k})^T \Sigma_{i,k}^{-1} (x_t - \mu_{i,k}))$
    with $\sum_{k=1}^{K} c_{i,k} = 1$

Hidden Markov models
three main problems

- How to compute the likelihood of an observation
  (for the speech recognition step)

- Given an observation, how to determine the state sequence which best explains the observation
  (for the speech recognition and for the training steps)
  ⇔ Viterbi algorithm

- Given a set of observations, how to determine the model parameters to best fit these observations
  (training)

note: observation ⇔ acoustic form ⇔ sequence of acoustic frame vectors

Training ⇔ Optimization of model parameters

- Determine the parameters of the model Λ in order to maximize the likelihood of the training data $X^* = \{X^1, X^2, \ldots, X^R\}$

$$\Lambda^{\text{opt}} = \underset{\Lambda}{\text{argmax}}\, \text{P}(X^*|\Lambda) = \underset{\Lambda}{\text{argmax}} \prod_r \text{P}(X^r|\Lambda)$$

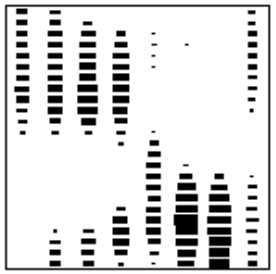
- Iterative approach
  - Initialization
    - Average value identical for all the pdf
      or more specific values using segmented and annotated speech data
  - Iterations – EM approach (*Expectation-Maximization*)
    - Estimation step (*expectation*) ➔ Estimate the probability of usage of the states, transitions, and Gaussian components when the model is used to generate the training data
    - Maximisation step ➔ Determine from the above estimates, new values of the model parameters (that increase the likelihood of the training data)
  - Stopping criterion
    - Convergence or maximum number of iterations

Remarks on the training process

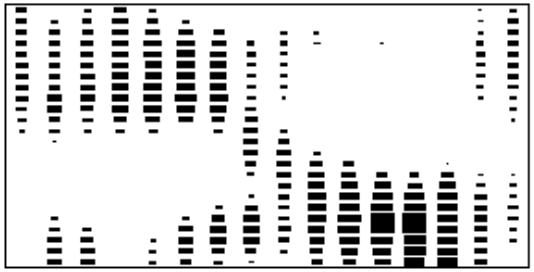
- Baum-Welch algorithm (i.e. above formula)
  - Takes into account all possible paths (state sequences)
  - Needs to take care of numerical problems (sums of probabilities)
    ➔ Computations (forward and backward variables) need to be normalized
- Viterbi algorithm
  - Handles only the information corresponding to the best matching path
  - Computations using logarithm of probabilities avoid numerical problems
- Both approaches converge toward a local extremum
- The trained model corresponds to the observations (speech data) of the training set. For optimal performance, the training set must match the foreseen usage conditions: speaker, environment, …
- Nevertheless, the choice of the topology (number of states, transitions, …) is to be done a priori

Examples of models for the French digit "0"

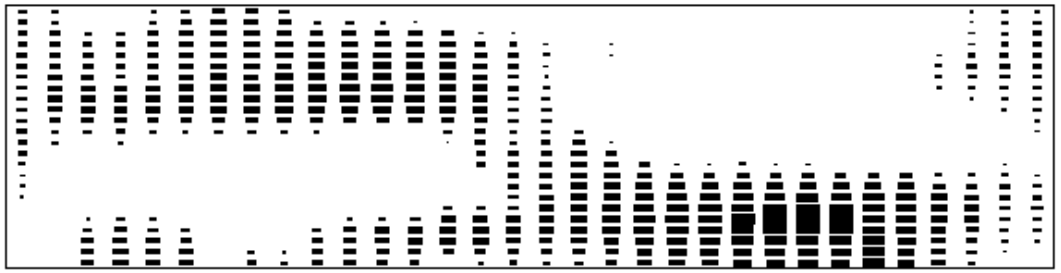
8 states model



16 states model



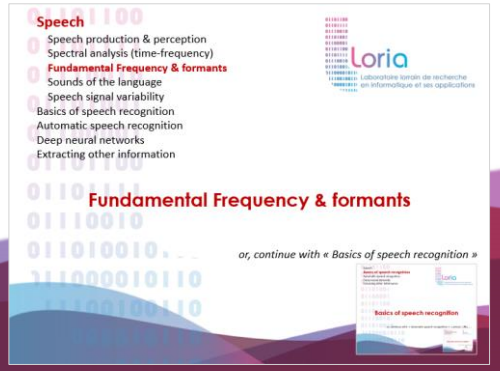
32 states model



Loria
Laboratoire lorrain de recherche en informatique et ses applications

Automatic speech recognition

or, back to « Speech » : F0, phones, variability, ….

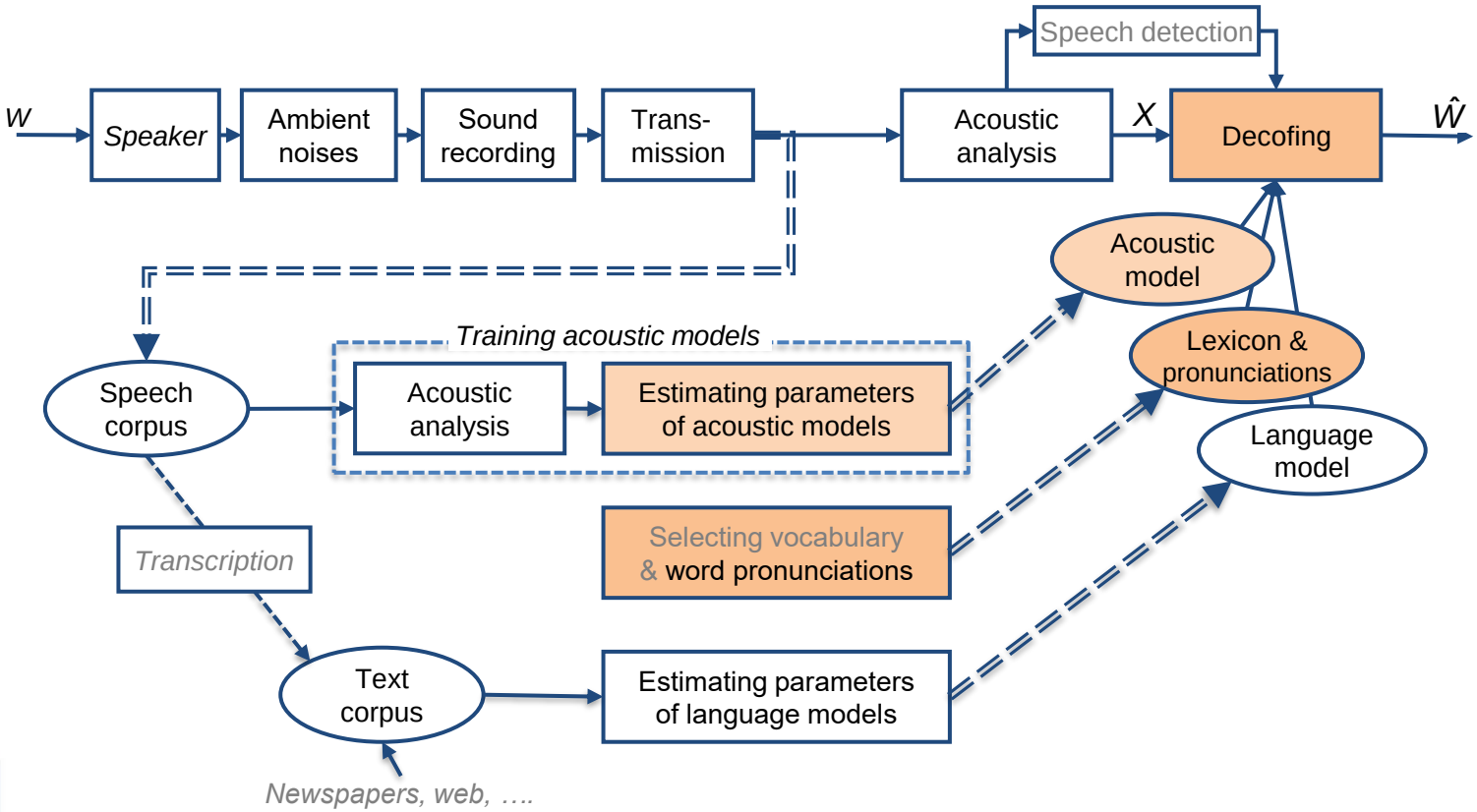


Loria
Laboratoire lorrain de recherche en informatique et ses applications

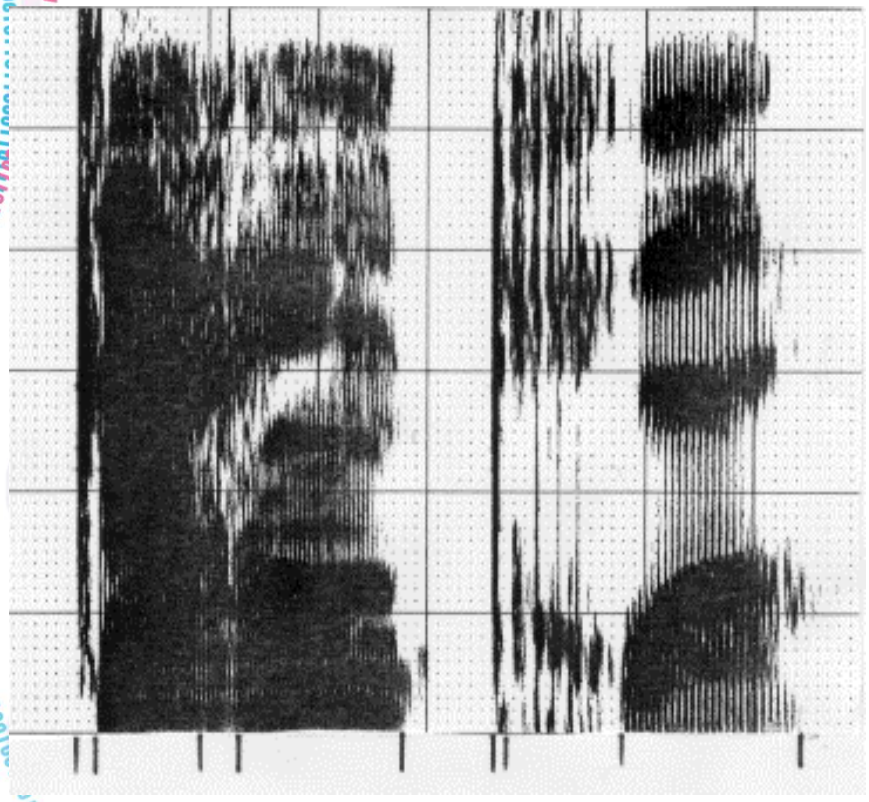
Lexicons and language models

Models of phones and pronunciation variants

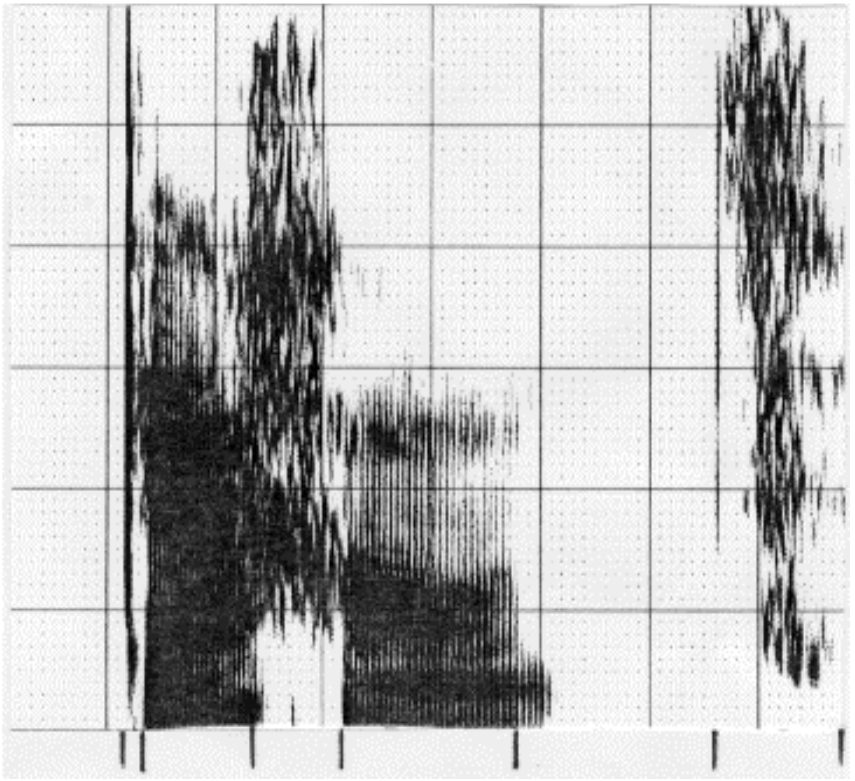
- Representation of the possible pronunciations of the words
  - Contextual modeling of the phones
  - Pronunciation variants



Allophonic variants



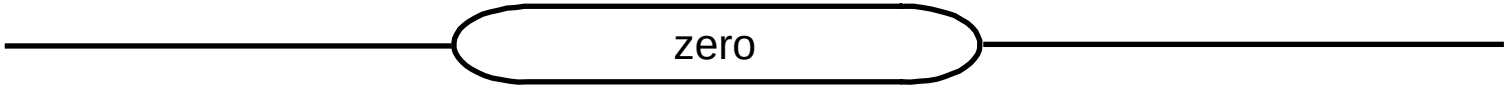
k a r ã t r w a



k a r ã t

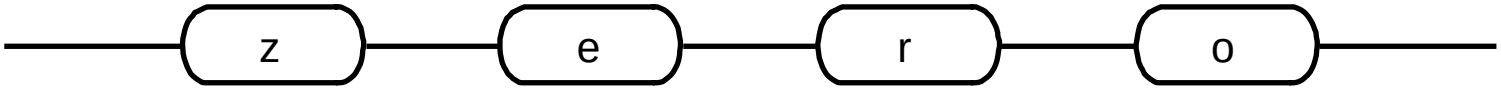
Modeling based on phonemes vs. words

Word-based



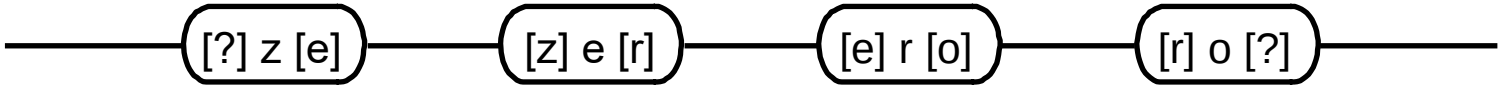
Robust for isolated words, but requires many examples of the words for training

Phoneme-based



Reduce number of units, language dependent
but modeling of contextual influence is not properly handled

Context-dependent phonemes

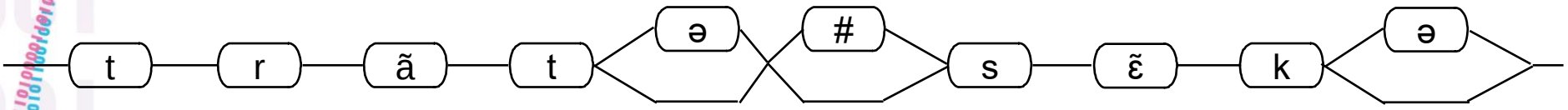


Number of possible units is very large ($N = 40$ phonemes => $N^3 \approx$ 64,000 triphones),
allows to model any word by concatenating the relevant models,
but problem for estimating some parameters ➔ sharing of parameters

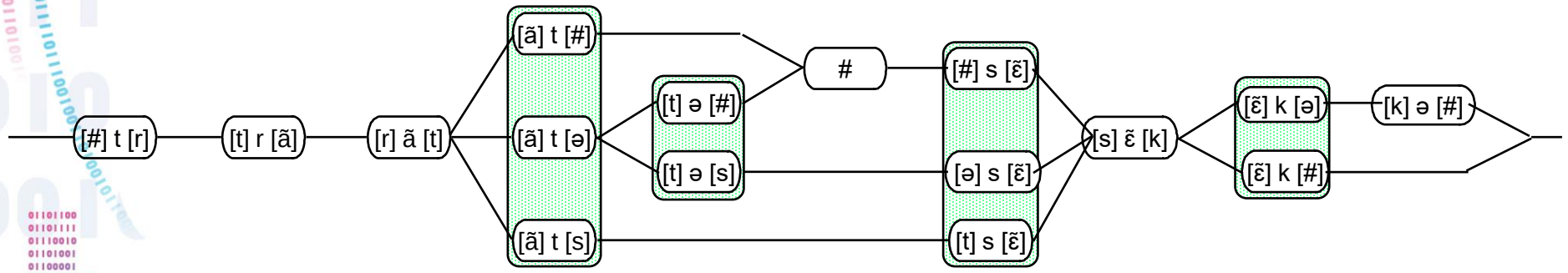
Modeling with context-dependent phonemes handling word boundary contexts

Example: possible pronunciations for French number "35"

Modelling with context-**independent** phonemes

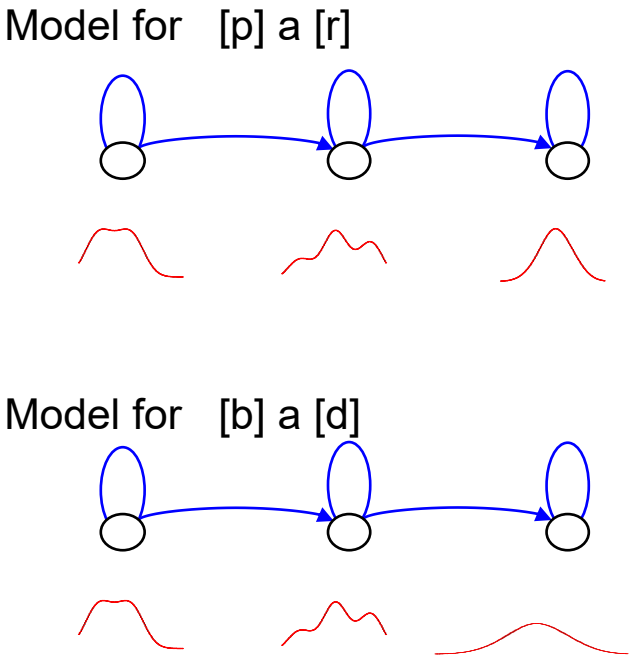


Modeling with context-**dependent** phonemes (called triphones)

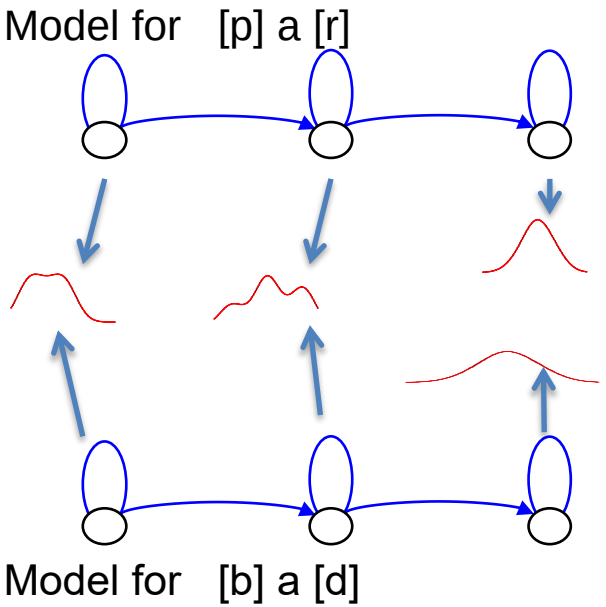


Sharing of pdfs for context-dependent phoneme modeling

pdf specific to each state



pdfs shared between states



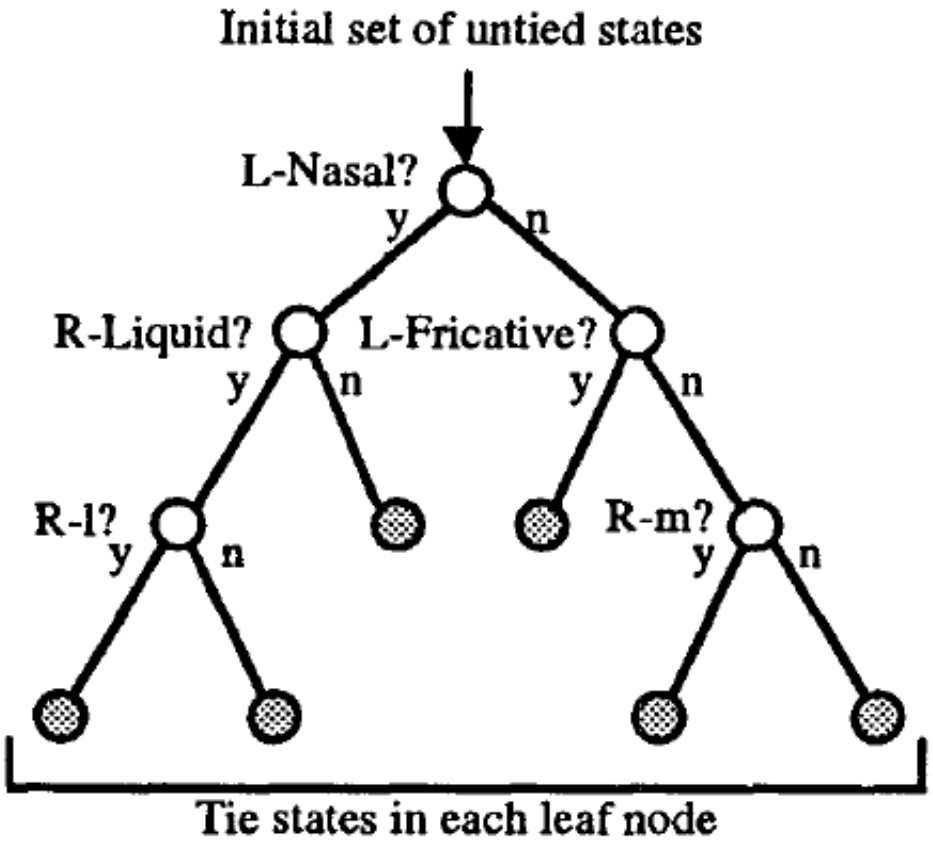
Sharing is determined automatically using decision tree

Sharing of pdfs for context-dependent phoneme modeling

Sharing determined automatically using decision trees

- Approach
  - Train a context-dependent phoneme model with **single Gaussian pdfs**
    ➔ 1 pdf (only one Gaussian component) for each state
  - Prepare a set of « questions »,
    each question corresponds to a class of phonemes, such as:
    « is-it a vowel? », « is-it a consonant? », « is-it a fricative? », …, « is it a voiced fricative? », ….
  - Decision trees are built through an iterative process
    that select for each node the question that leads to the best score (i.e., highest probability for the resulting pair of data subsets)
  - Then, arbitrary choice of the total number of shared pdfs

Example of decision tree



- In practice, the resulting decision trees are much deeper than on this example

Lexicon – pronunciation variants

- Associate to each word of the lexicon, the possible pronunciation variants, e.g.:

très	t R e
très(2)	t R e z
verte	v e R t
verte(2)	v e R t swa

vertes	v e R t
vertes(2)	v e R t swa
vertes(3)	v e R t z
vertes(4)	v e R t swa z

- In French, the largest amount of pronunciation variants corresponds to
  - To the pronunciation, or not, of the schwa /ə/ at the end of the word
  - To the pronunciation of a liaison consonant, after some words, when the next one starts by a vowels

- Examples in English (from cmudict)

soften	S AA F AH N
soften(2)	S AO F AH N

sorbet	S AO R B EY
sorbet(2)	S AO R B EH T

Pronunciation variants

- How to obtain the pronunciations variants of the words?
- From pronunciation lexicons (when available)
- Through the use of graphemes-to-phonemes converters
  - Rules (defined manually)
  - Automatic approaches
    - *Joint Multigram Models*,
    - *Conditional Random Fields*,
    - *Neural networks, …*

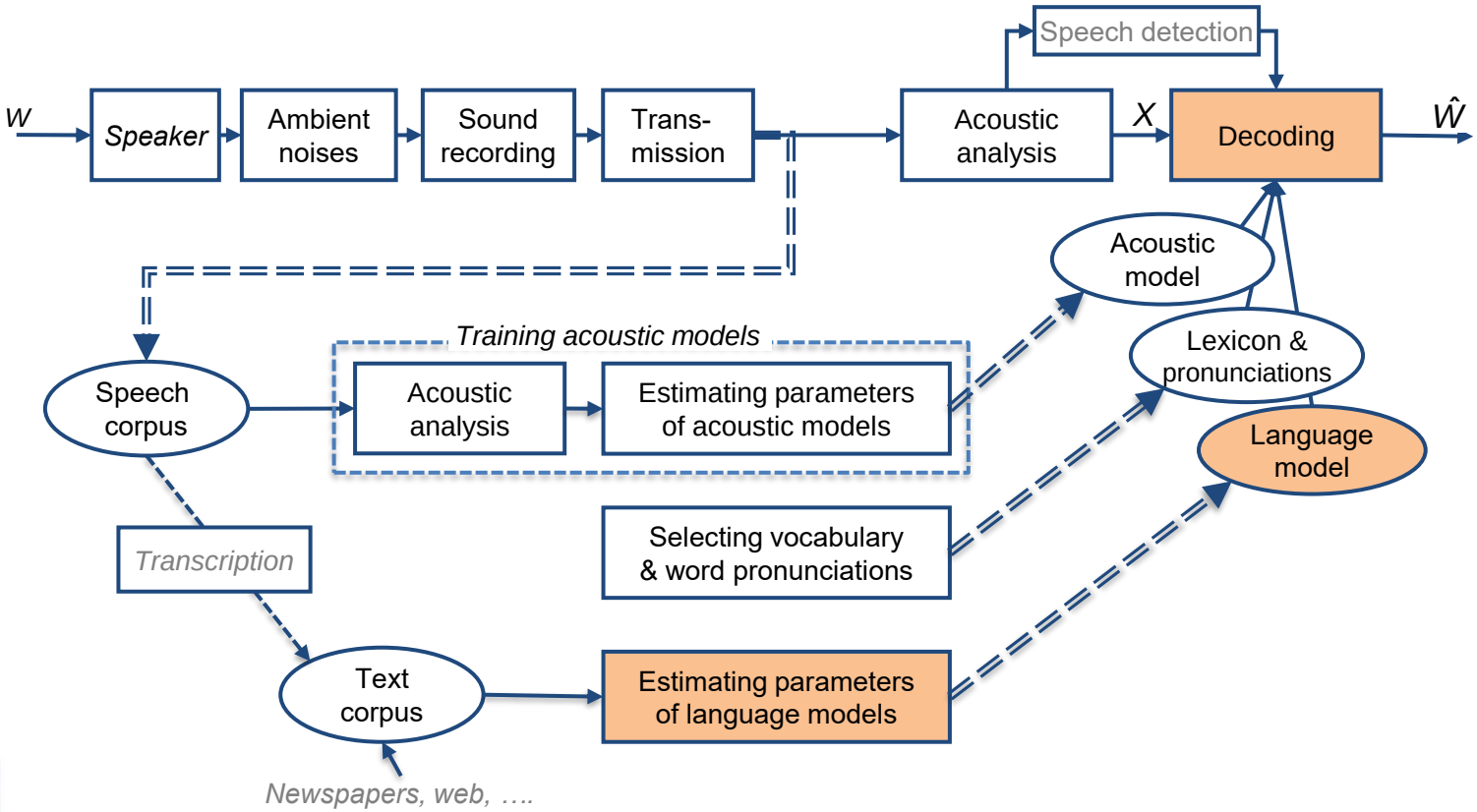
    trained from data (large set of examples < word ⇔ pronunciations >)

Lexicons

- The lexicon is critical
- The lexicon specifies the list of words known by the ASR system
  - An ASR system cannot recognized words that are not in the lexicon (words not in the lexicon are also called “out-of-vocabulary words”)
  - It is impossible to have a lexicon covering all possible words (because of person names, company names, product names, …)
  - Diachronic evolution of vocabularies (e.g., due to new topics in the news that refer to new persons, new locations, …)
- The lexicon also specifies the possible pronunciations of the words
  - Must include the usual pronunciation variants
  - But one should not include too many useless variants as this increases possible confusions between vocabulary words (because of that, the recognition of non-native speech is still a problem)

Language models

- Representation of the possible sequences of words
  - Modeling
  - Usage in speech decoding



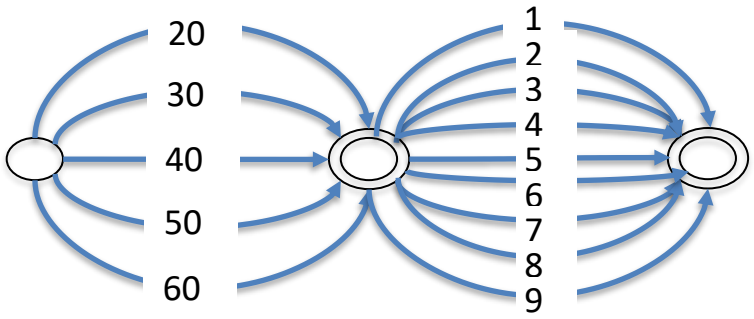
Language models

- Provide information on the possible sequences of words
- Context-free grammars
  - Specify sequences of words corresponding to « sentences » (i.e. global constraints)
- n-gram statistical model
  - Provide local constraints (on sequences of *n* words)
  - Estimation from text corpora
- Neural network models
  - Many different approaches proposed in the literature

Context-free grammars

- Rules (or graphs) that describe exactly the allowed sentences (sequences of words) of the language (e.g., isolated digits, numbers, …)
  - Ex. Number_from_20_to_69

```
Number_from_20_to_69 =  Tens_from_20_to_60
                     |  Tens_from_20_to_60 . Units_from_1_to_9 ;
Tens_from_20_to_60   =  20 | 30 | 40 | 50 | 60 ;
Units_from_1_to_9    =  1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ;
```



- Global constraints on the sentences
- Complex and difficult to define for large vocabularies
- Do not allow the recognition of sentences that do not respect the grammar

Example of grammar

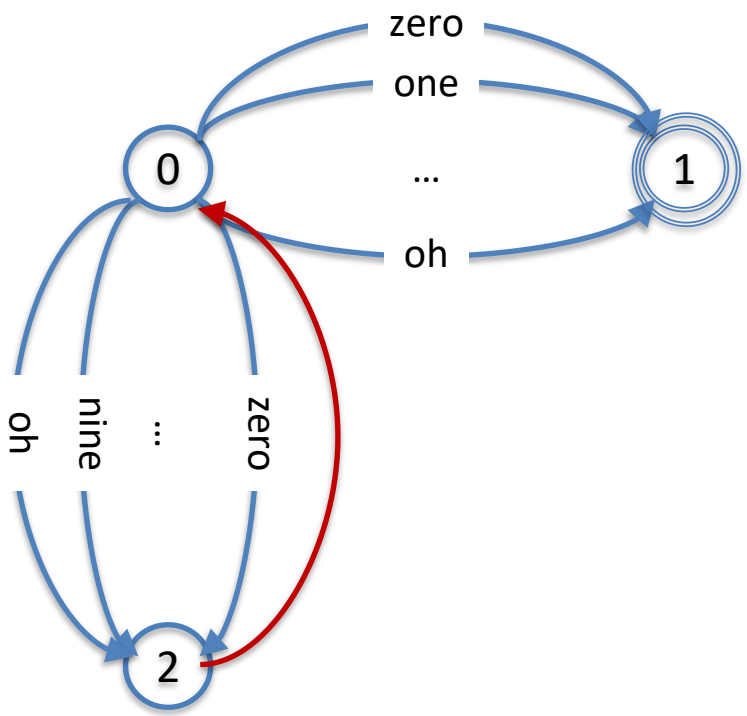
- Grammar JSGF – digit loop

```
#JSGF V1.0
grammar digits;
public <digitloop> = <digit> + ;
<digit> = zero | one | two | three | four | five | six | seven | eight
| nine | oh ;
```

- Excerpt FSG

```
FSG_BEGIN <digits.digitloop>
NUM_STATES 3
START_STATE 0
FINAL_STATE 1
TRANSITION 0 1 1.000000 zero
TRANSITION 0 1 1.000000 one
……
TRANSITION 0 1 1.000000 oh
TRANSITION 0 2 1.000000 zero
……
TRANSITION 0 2 1.000000 nine
TRANSITION 0 2 1.000000 oh

TRANSITION 2 0 1.000000
FSG_END
```



N-gram language model

- Statistical model
- Probability of a word sequence

$$P(w_1, \dots, w_N) = P(w_1) \prod_{i=2..N} P(w_i | w_1, \dots, w_{i-1})$$

- N-gram approximation, as n-grams deal with sequences of n words

$$P(w_i | w_1, \dots, w_{i-1}) \triangleq P\left(w_i \middle| w_{i-(n-1)}, \dots, w_{i-1}\right)$$

- N-gram parameters are computed from large text corpora

Examples of n-grams

- Sentence : *the sky is blue*

  ➔ *<s> the sky is blue </s>* including start and end of sentence symbols

- Unigrams

  Counts on text training corpus

  P(Sentence) = P(*the*) . P(*sky*) . P(*is*) . P(*blue*)

  $$\mathrm{P}(\text{Sentence}) = \prod_i \mathrm{P}(w_i) \qquad \mathrm{P}(w_i) = \frac{\text{count}(w_i)}{\sum_j \text{count}(w_j)} \qquad \mathrm{P}('est') = \frac{\text{count}('is')}{\sum_j \text{count}(w_j)}$$

- Bigrams

  P(Sentence) = P(*the*|<s>) . P(*sky*|*the*) . P(*is*|*sky*) . P(*blue*|*is*) . P(*</s>*|*blue*)

  $$\mathrm{P}(\text{Sentence}) = \mathrm{P}(w_1 | < s >) \prod_{i=2,..N} \mathrm{P}(w_i | w_{i-1})$$

  $$\mathrm{P}(w_i | w_{i-1}) = \frac{\text{count}(w_{i-1} w_i)}{\sum_j \text{count}(w_{i-1} w_j)} \qquad \mathrm{P}('is' | 'sky') = \frac{\text{count}('sky'\ 'is')}{\sum_j \text{count}('sky'\ w_j)}$$

- Trigrams

  P(Sentence) = P(*the*|<s>) . P(*sky*|<s> *the*) . P(*is*|*the sky*) . P(*blue*|*sky is*) . P(*</s>*|*is blue*)

n-gram language model – bigrams

- Bigrams ⇔ sequences of 2 words ➔ $\mathrm{P}(w_i|w_{i-1})$
- Estimation

$$\mathrm{P}(w_i|w_{i-1}) = \frac{\mathrm{N}(w_{i-1},w_i)}{\mathrm{N}(w_{i-1})} \qquad \text{if } \mathrm{N}(\mathrm{w_{i-1}}) \text{ non zero}$$

- N-gram provide local constraints (bigrams ⇔ sequences of 2 words)
- When the speaker does not follow the expected syntax, speech recognition errors are usually limited to a few words
- For a 1,000 word vocabulary, there 1,000,000 possible 2-word sequences!
- It is impossible to estimate exactly all the parameters, so “smoothing” techniques are used (classes of words, backoff, ...)

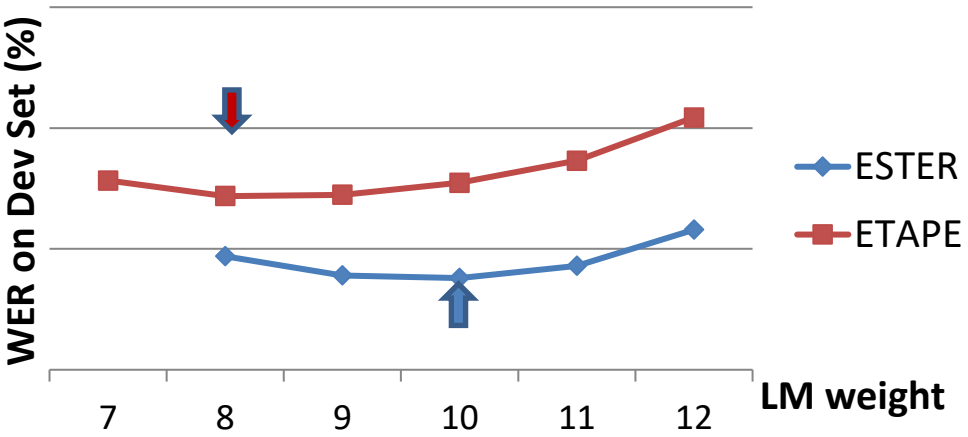
Weighting the language model contribution

- Decoding

$$\widehat{W} = \underset{W}{\operatorname{argmax}}\, \mathrm{P}(X|W)\mathrm{P}(W)$$

- Weighting the language model contribution (*fudge factor*) : γ

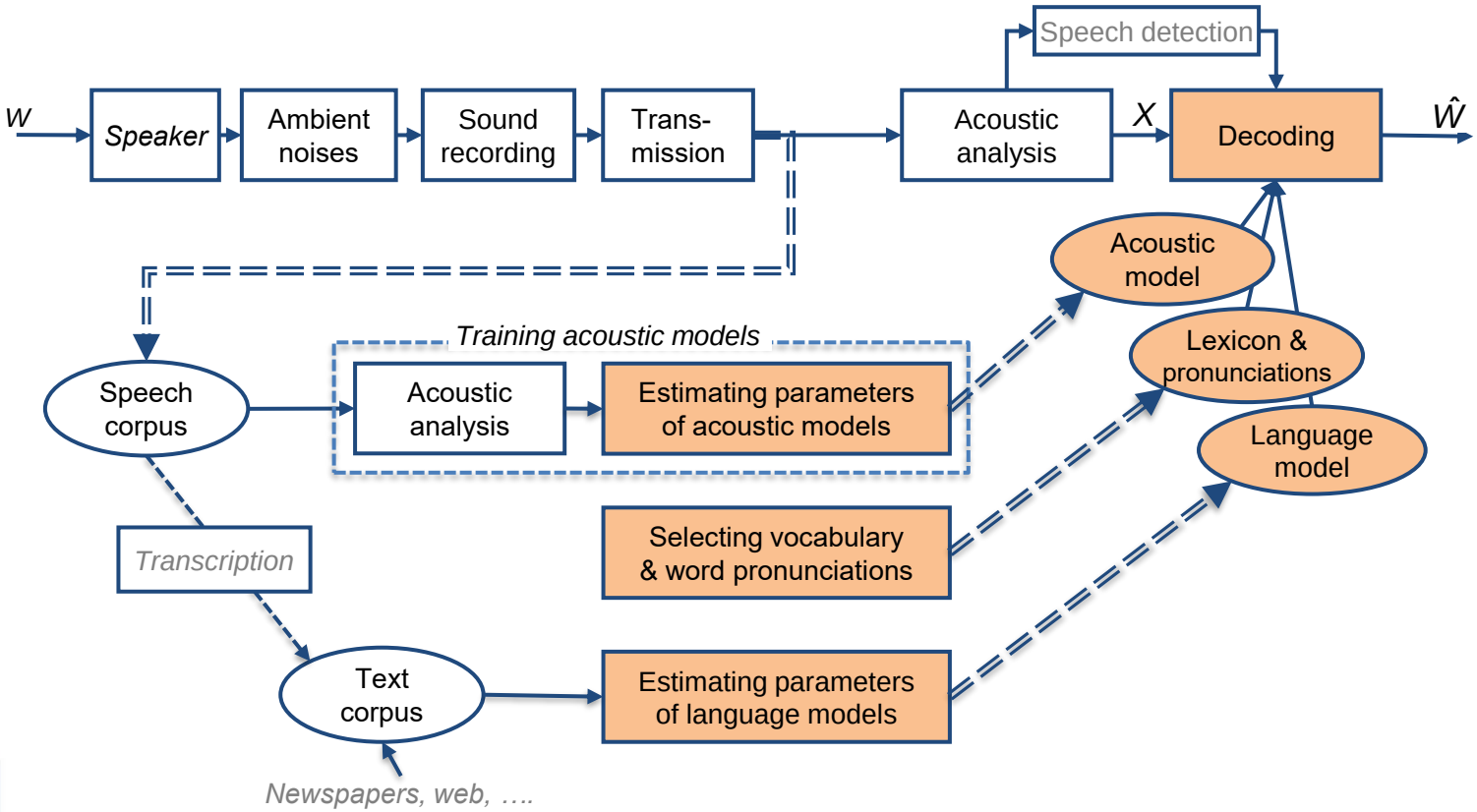
$$\widehat{W} = \underset{W}{\operatorname{argmax}}\, \mathrm{P}(X|W)\big(\mathrm{P}(W)\big)^{\gamma}$$



- Optimal weighting depends on the quality and relevance for test conditions, of the acoustic and language models

Large vocabulary speech recognition

- Main steps for training models
- Heuristics for decoding



Building context-dependent phoneme models

- Relies on a transcribed speech corpus
  [transcriptions ⇔ sequences of words]

- First step using the "standard pronunciation" of the words
  [➔ which thus defines the sequence of phone HMMs corresponding to each utterance]
  1. Training a simple model: context-independent phoneme models (~35 models) and single Gaussian component densities
  2. Training context-dependent phoneme models (single Gaussian component densities, and no sharing) – *initialized from the context-independent phoneme models*
  3. Determine the sharing of the pdfs between the states (*decision trees*)
  4. Training context-dependent phoneme models (with shared single Gaussian component densities) – *initialized from the context-independent phoneme models*
  5. Then continue training while increasing the number of Gaussian components per density (2, 4, 8, …)

- Second step
  – Forced alignment (text ⇔ speech) of the training speech corpus (➔ this determine the pronunciation variant associated to each word occurrence)
  – Redo the training of the model (sub steps 1 to 5 above) but using the pronunciation variant associated to each word occurrence (resulting from the forced alignment)

Example of sizes of training corpora and models

For speech transcription using hidden Markov models (*in years around 2010*)

- Lexicon: 100 000 mots (or more)
- Text data for training language models
  - about 1,6 Giga words from various sources
    newspapers (500 M words), broadcast news transcriptions (100 M words), web data (250 M words), corpus Gigaword (750 M words)
- Transcribed speech data for training acoustic models
  - about 300 hours (⇔ 100 millions of frames)
- Typical size of acoustic models
  - 7500 shared densities
  - 64 Gaussian components per density
  - 39 acoustic features per frame (Energy, 13 MFCC, plus derivatives)
    * 2 (mean & variances)

  => Hence a total of about 38 millions parameters

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Measures of performance

Speech recognition errors (isolated words – vocal commands)

Possible answers

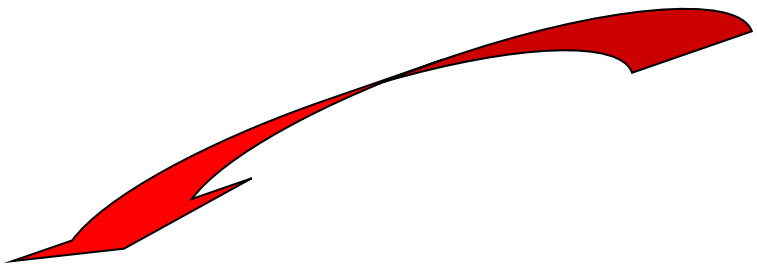
⇔ vocabulary known by the ASR system

« previous »

« next »

« back »

« cancel »



Uttered word (by speaker)	*Recognized word (by ASR system)*	***Type of error***
« next »	« cancel »	Substitution
« erase » *(out-of-vocabulary word)*	« back »	False alarm
-- *(noise)*	« next »	False alarm
« back »	-- *(i.e. rejected)*	False reject

Speech recognition errors continuous speech

- **Substitution**
  - Ref. : **Lannion**
  - Reco. : **Lyon**

- **Insertion / Deletion / Substitution**
  - Ref. : **I want to go to Lyon**
  - Reco. : **well I want to go Lannion**

Word error rates

- Estimated after aligning recognized words (word hypotheses) with reference transcription (manual annotation)

- Example of alignment result
  - REF : reference
  - HYP : results of speech recognition system

```
Start-time: 382.484
End-time: 395.969
Scores: (#C #S #D #I) 40 0 1 2
REF:  la source de la contamination **** **** n' a ELLE pas encore été trouvée
HYP:  la source de la contamination MAIS ELLE n' a **** pas encore été trouvée
Eval:                               I    I         D
>> REF:  ce bilan d' une réunion d' experts des vingt sept états de l' Union
>> HYP:  ce bilan d' une réunion d' experts des vingt sept états de l' union
>> Eval:
>> REF:  Européenne contraste avec les soupçons portés par les autorités
>> HYP:  européenne contraste avec les soupçons portés par les autorités
>> Eval:
>> REF:  sanitaires allemandes sur des concombres espagnols
>> HYP:  sanitaires allemandes sur des concombres espagnols
>> Eval:
```

Word error rate

- WER : Word Error Rate

$$WER = \frac{N_{sub} + N_{ins} + N_{del}}{N_{refwords}}$$

- **Confidence intervals**

  ⇔ Measure the uncertainty on the estimated word error rate

  95% confidence interval ➔ $1.96\sqrt{\frac{P\,.(1-P)}{N}}$

  where P is the word error rate, and N the number of words in the test set

  for example

  for a test set of 4000 words, and a word error rate of 8%

  ➔ $1.96\sqrt{\frac{0.08\,.(1-0.08)}{4000}} = 0.008$

  which give a word error rate of 8.0%±0.8%

Typical speech recognition output

- Example of English speech recognition output

id_of_audio_file	canal	t_start	length	word	confidence
Euronews_eng_AVmedium_zx8MkqjsC3s	A	1.77	0.34	donald	0.77
Euronews_eng_AVmedium_zx8MkqjsC3s	A	2.14	0.38	trump	0.85
Euronews_eng_AVmedium_zx8MkqjsC3s	A	2.58	0.18	is	0.83
Euronews_eng_AVmedium_zx8MkqjsC3s	A	2.76	0.44	threatening	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	3.20	0.12	to	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	3.32	0.25	sue	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	3.57	0.09	the	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	3.66	0.19	new	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	3.85	0.24	york	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	4.09	0.53	times	0.96
Euronews_eng_AVmedium_zx8MkqjsC3s	A	4.62	0.42	following	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	5.04	0.12	an	0.99
Euronews_eng_AVmedium_zx8MkqjsC3s	A	5.16	0.58	article	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	5.74	0.47	alleging	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	6.21	0.19	he	1.00
Euronews_eng_AVmedium_zx8MkqjsC3s	A	6.40	0.17	had	1.00

Confidence measure and rejection

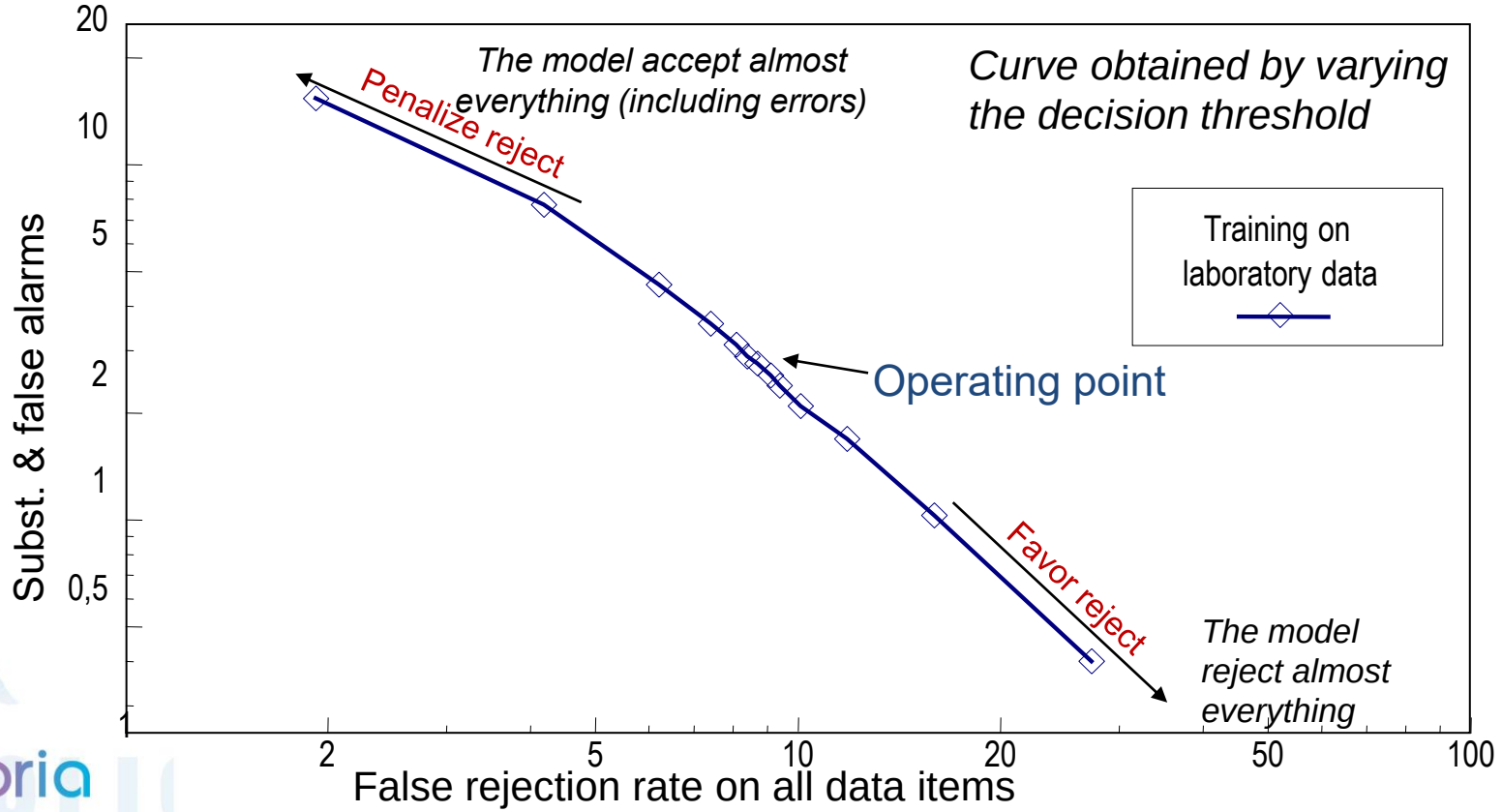
In vocal interaction context

- Comparison of the confidence measure to a decision threshold
  - Above the threshold
    - The speech recognition output will be sent and processed by the dialog management system
  - Below the threshold
    - The recognized word is considered as « unreliable » and is thus not processed further

    ➔ Ask for a repetition of the command…

Setting the decision threshold

- Adjust the trade-off between
  - false rejection *(the system reject a word, although it belongs to the ASR vocabulary)*
  - false alarm *(for a noise or an out-of-vocabulary word, the system recognize it as a vocabulary word, but does not reject the answer)*



Impact of a mismatch between training and test data

- WER with respect to telephone network mismatch

		Model (training data)		
		PSTN	*GSM*	*PSTN & GSM*
Test data	*PSTN*	0,8%	1,9%	1,0%
	GSM	4,7%	1,9%	2,2%

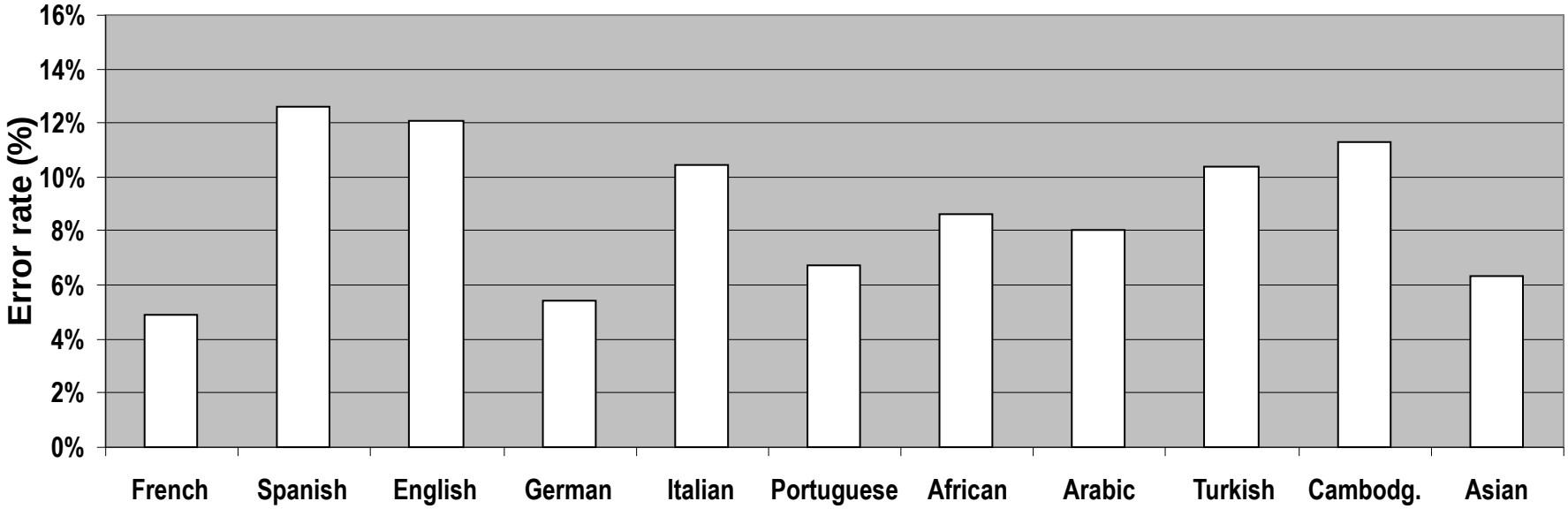
(digit corpus)

- WER with respect to gender mismatch

		Model (training data)	
		Men	*Women*
Test data	*Men*	10,7%	26,9%
	Women	16,3%	8,1%

Performance on non-native speech

- 700 speakers from 24 countries, uttering 83 French words/expressions
- Standard French modeling (native French acoustic models, native pronunciations variants)



Speech recognition performance vary a lot between speaker groups

(Error rate: from less than 6% for German … up to more than 12% for English & Spanish)

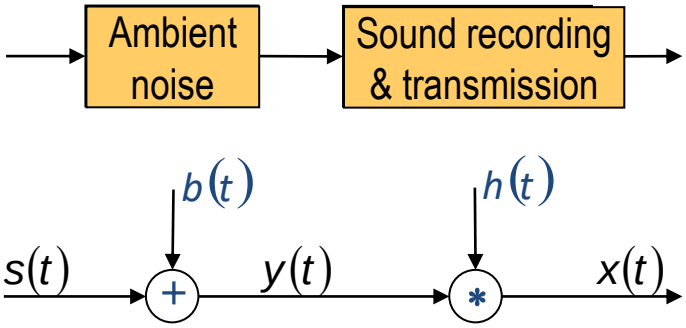
Dealing with speech signal variability

- Adapting acoustic features
  - Speech enhancement (noise estimation & suppression – *signal processing*)
- Adapting acoustic models
  - Adapting a generic model to specific (new) conditions using small amounts of speech data
- Class-based modeling
  - Building several acoustic models, each one being associated to a class of data. At decoding, first estimation of the class id, then decoding with the associated acoustic model
- Multi-condition training
  - Use speech data from various conditions to train a single acoustic model.
  - Possibly use additional artificially corrupted speech data for training

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Robustness & adaptation

Robustness with respect to environment



- At signal level

$$x(t) = y(t) * h(t)$$
$$= \big(s(t) + b(t)\big) * h(t)$$

- At spectral level

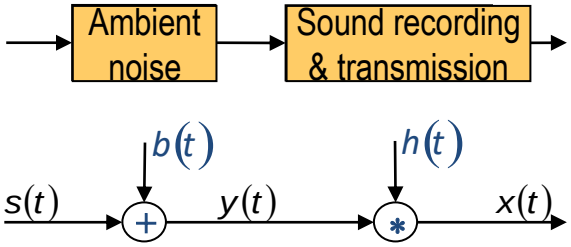
$$X(f) = \big(S(f) + B(f)\big).H(f)$$
$$= S(f).H(f) + B(f).H(f)$$

Clean signal

Impact of sound recording & transmission

Impact of noise

Robustness with respect to environment processing additive noise



- At spectral level

$$X(f) = S(f).H(f) + B(f).H(f)$$

- Thus

$$S(f).H(f) = X(f) - B(f).H(f)$$

Contribution from clean signal

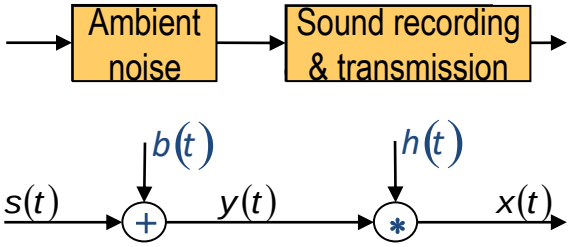
Noisy signal

Contribution of the noise

➔ Estimation of the noise spectrum on non-speech portions, then subtraction (assuming noise is stationary!)

⇔ Spectral subtraction

Robustness with respect to environment processing convolutive noise



- At spectral level

$$X(f) = \big(S(f) + B(f)\big).H(f)$$

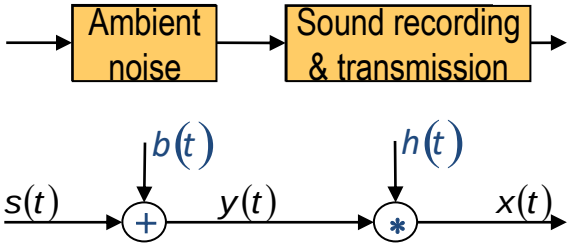
- Thus

$$\log\big(X(f)\big) = \log\big(S(f) + B(f)\big) + \log\big(H(f)\big)$$
$$\approx \log\big(S(f)\big) + \log\big(H(f)\big) \qquad \text{if } |S(f)| \gg |B(f)|$$

- Hence $C_X(n) = C_s(n) + C_h(n)$

- Impact of convolutive noise (sound recording & transmission)
  ⇔ Additive bias on the cepstral coefficients, so
  - Estimation of the bias ($C_h(n)$),
  - And, then subtraction of the bias (⇔ cepstral subtraction)

Robustness with respect to environment



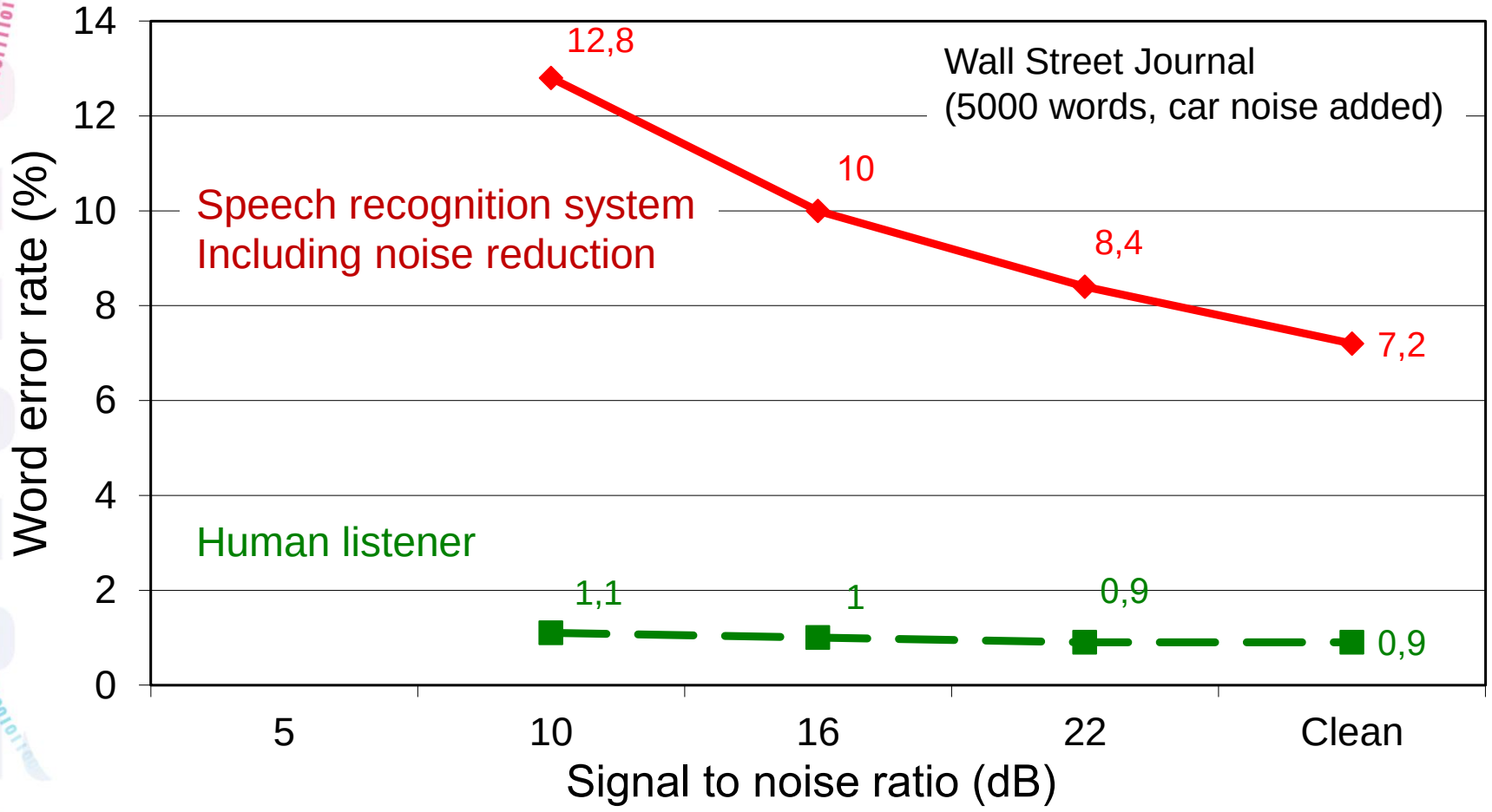
- Processing additive noises
  - Spectral subtraction, speech enhancement
- Processing convolutive noises (sound recording & transmission)
  - Cepstral subtraction, de-reverberation
- Noise robust acoustic analysis
  - Acoustic analysis which include specific processing for dealing with additive noises and convolutive noises
- Hands-free distant (multi-)microphone
  - Using a microphone array allows to apply more efficient enhancement approaches
- And also, more recent enhancement techniques relying on neural networks

Source separation and robust speech recognition

Examples of speech enhancement and impact on speech recognition performance

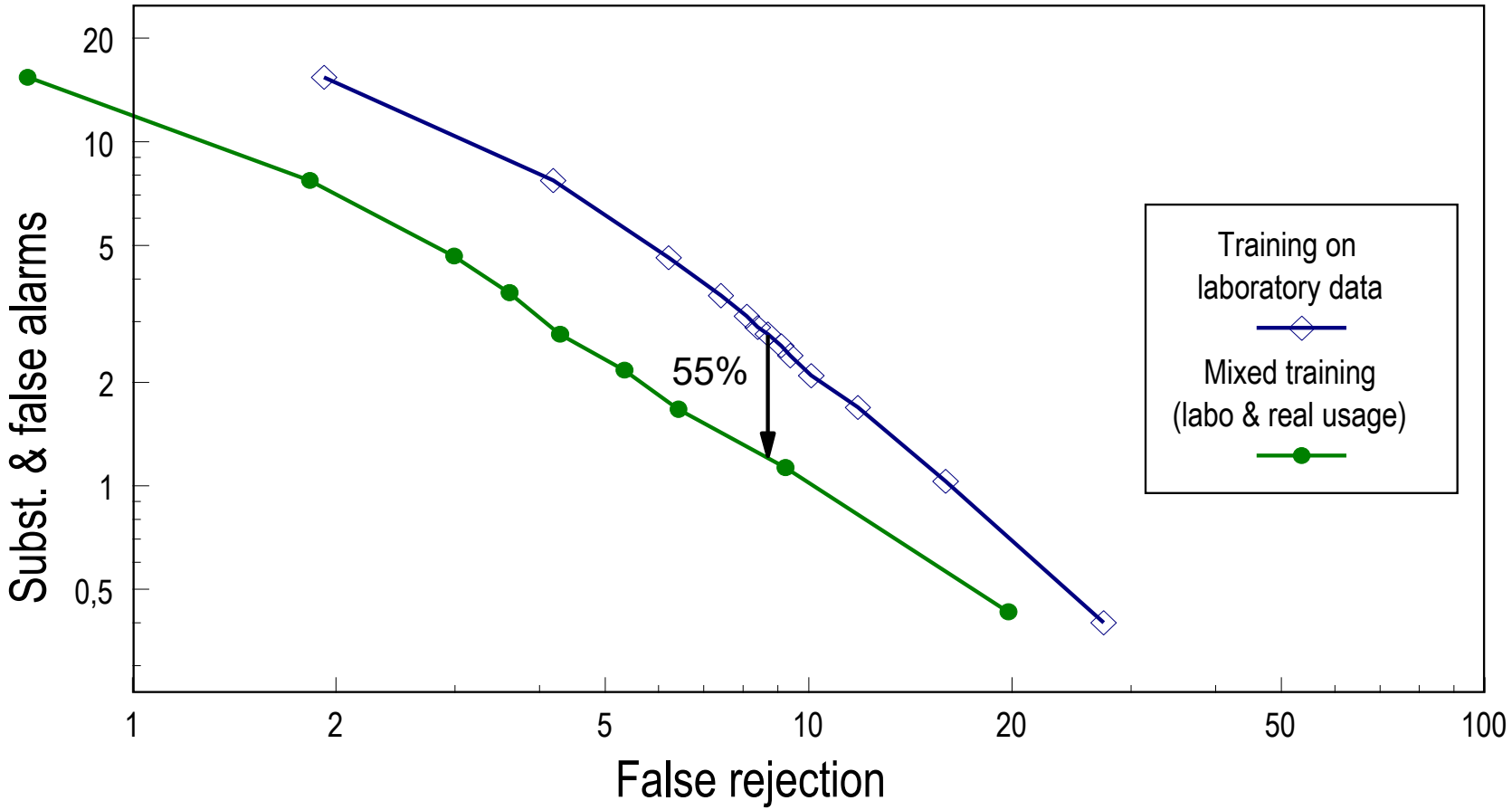
- Recorded signal   (baseline)
- Deep learning single channel (i.e., use a single microphone)   no WER reduction
- Beamforming (multiple microphones)   21% relative WER reduction
- Deep learning multi-channel   39% relative WER reduction

Impact of noise on speech recognition performance



Adaptation on real usage data (example)

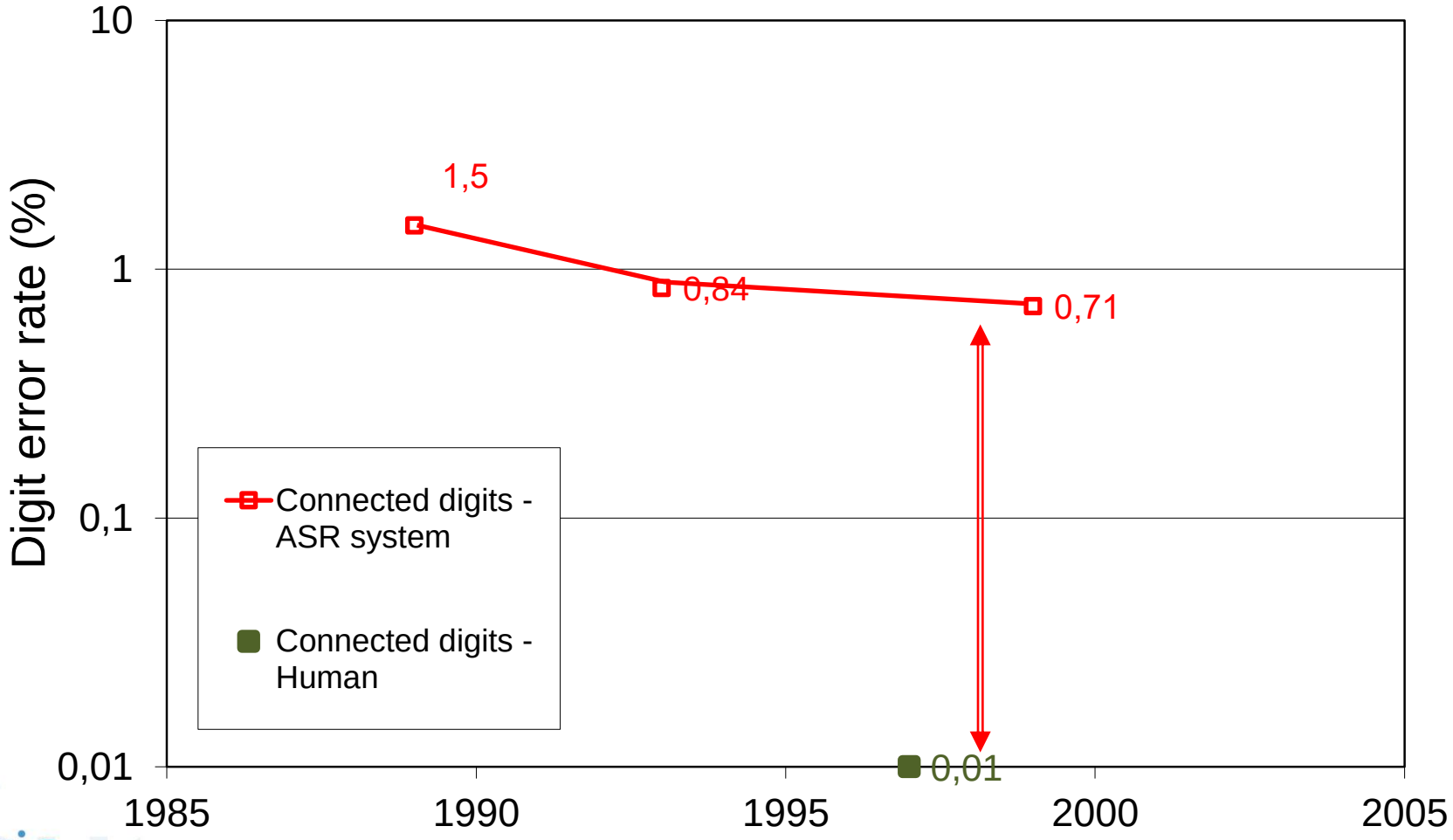
curves obtained by modifying the rejection threshold



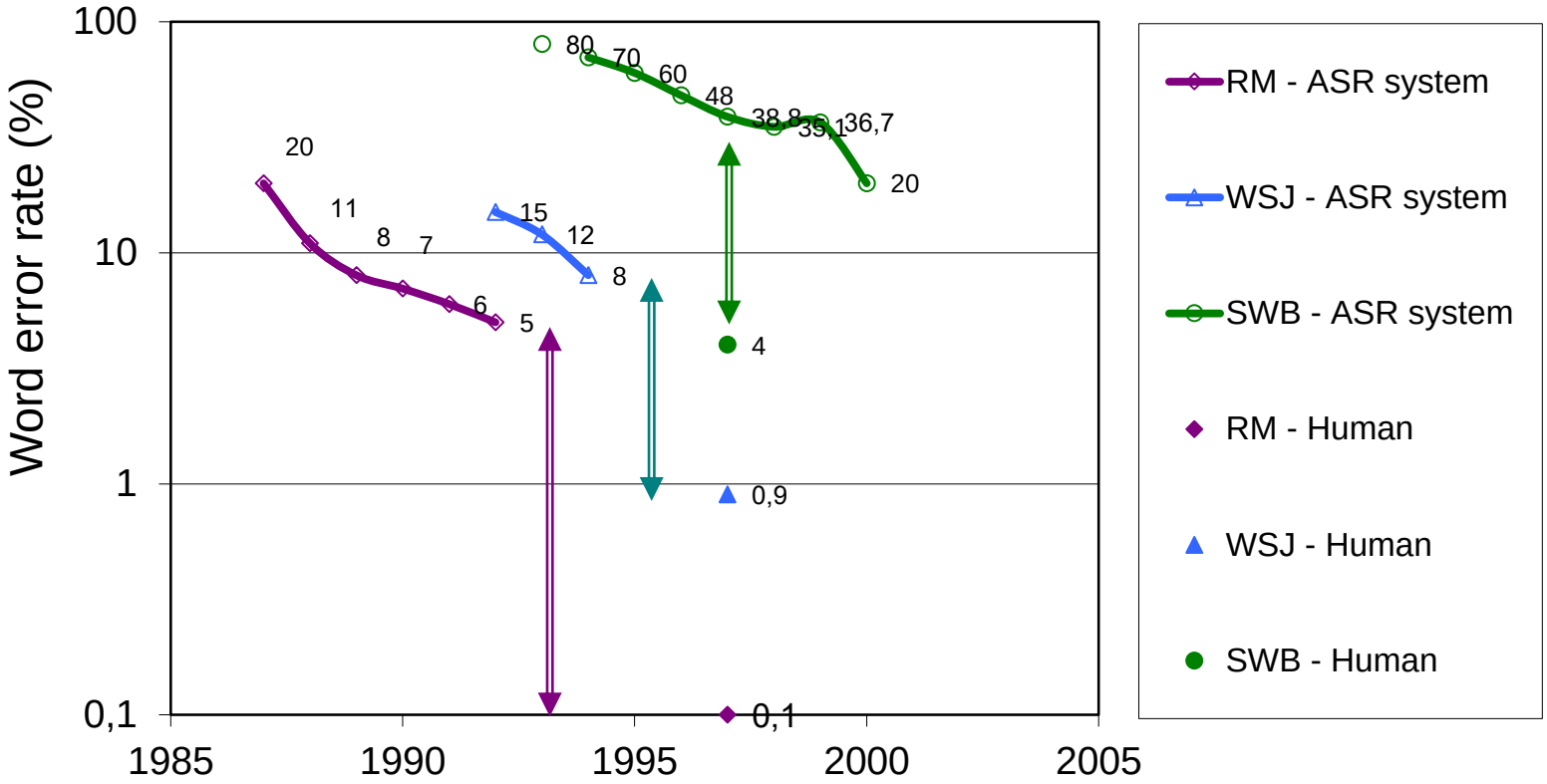
Loria
Laboratoire lorrain de recherche en informatique et ses applications

Performance improvements

Performance improvement over time on connected digit data

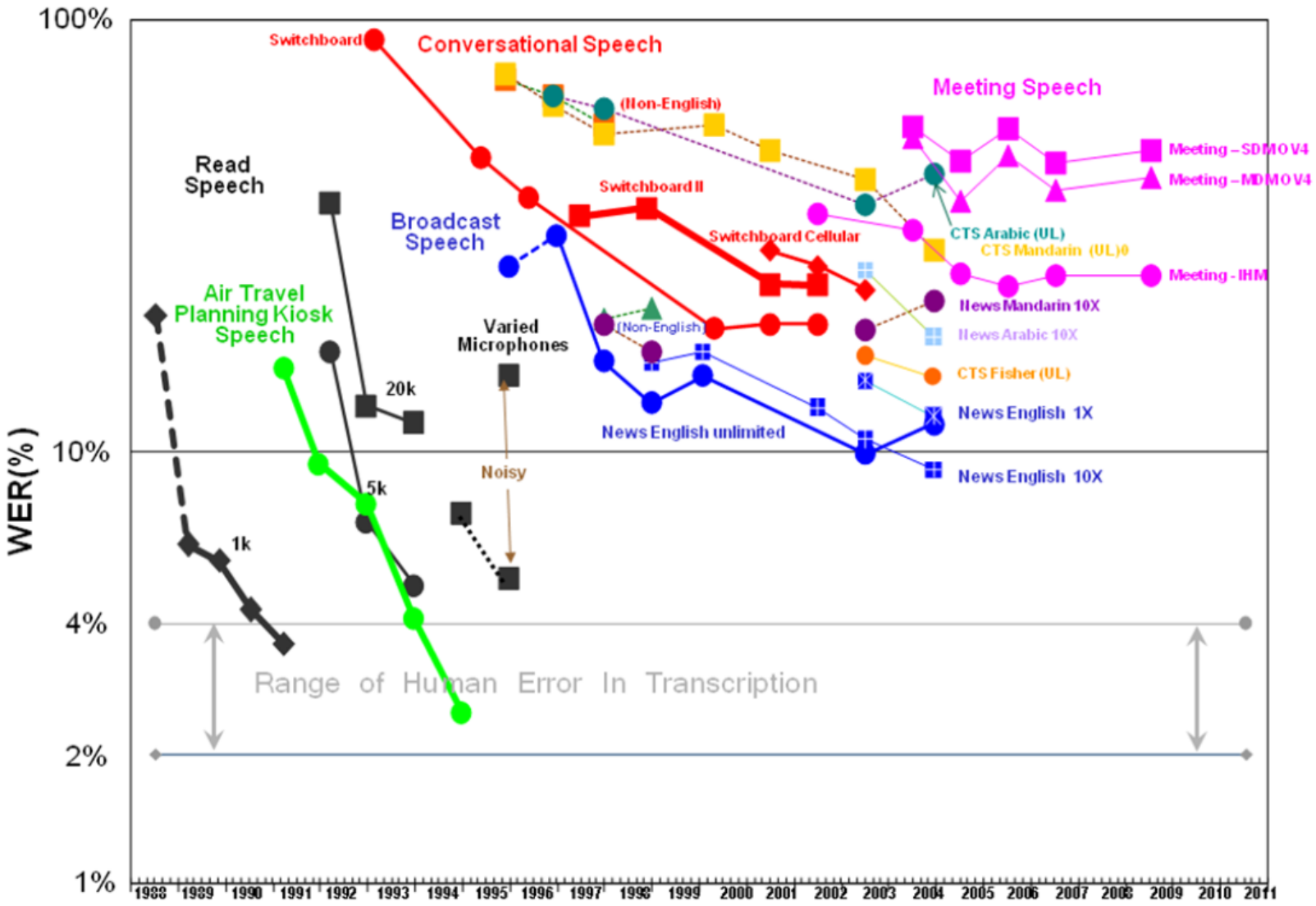


Performance improvement over time



RM *(Ressource Management)*:	Read sentences, clean data,	1000 words.
WSJ *(Wall Street Journal)*:	Read sentences, clean data,	5000 words.
SWB *(Switchboard)*:	Telephone conversations,	28000 words.

NIST speech transcription benchmark test history – May 2009



Loria
Laboratoire lorrain de recherche en informatique et ses applications

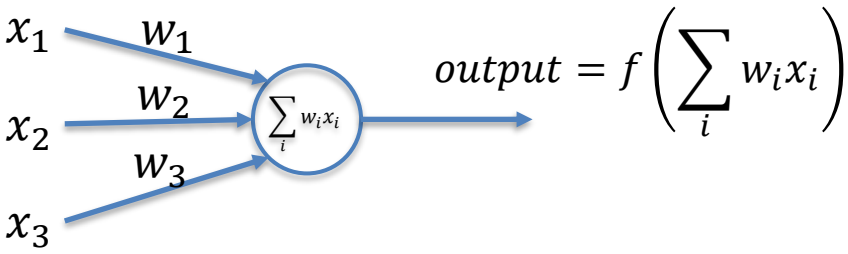
Deep neural networks

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Neural networks

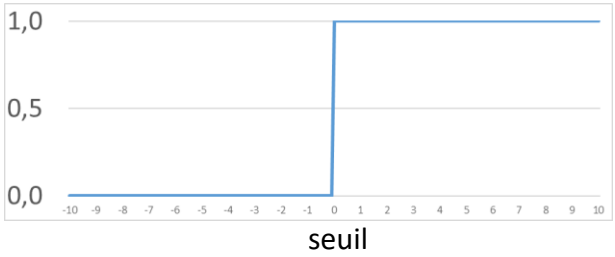
Perceptron

- Compute
  - Weighted sum of input values
  - Then, apply a non linear function



- Example of non linear function
  - Threshold

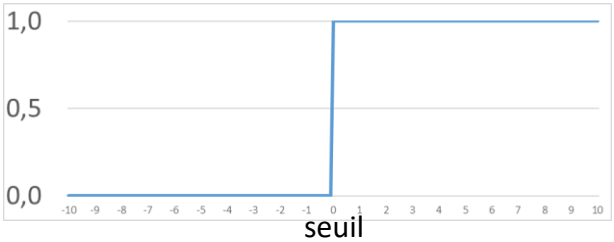
$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{else} \end{cases}$$



Example of non linear functions

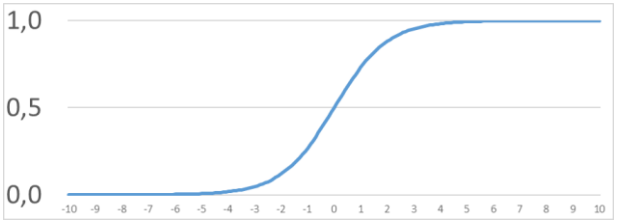
- Threshold

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{else} \end{cases}$$



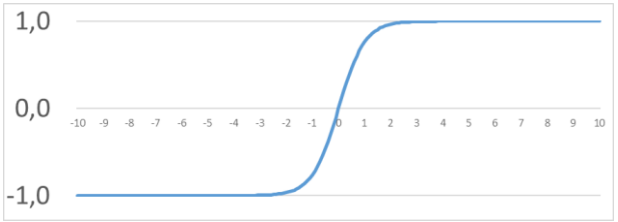
- Sigmoid

$$f(x) = \frac{1}{1+e^{-x}}$$



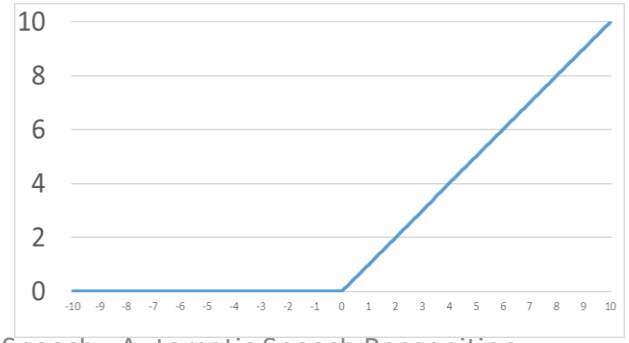
- Hyperbolic tangent

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$



- ReLU (rectified linear unit)

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$



Offset (threshold)

- The cell output y_j is given by

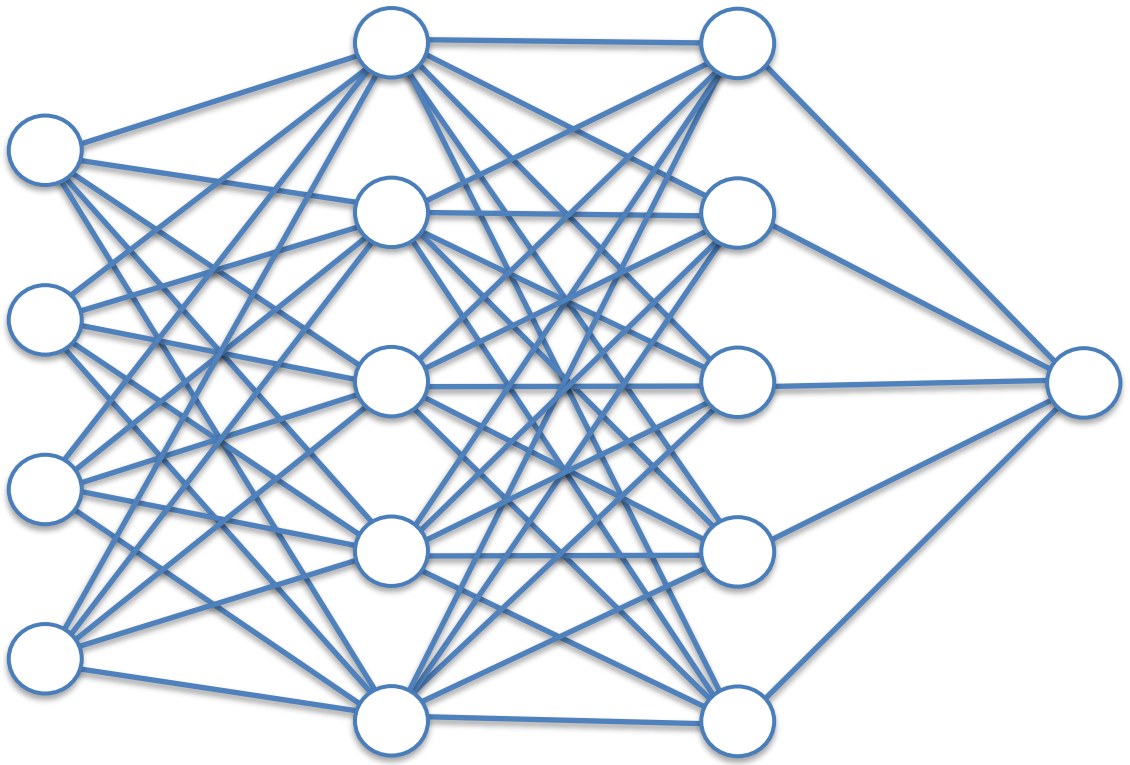
$$y_j = f\left(\sum_i w_{ij} x_i\right)$$

- Which corresponds to a threshold equal to 0, for the non linear functions

- To set another threshold, a bias (or offset) b_j is added in the computations

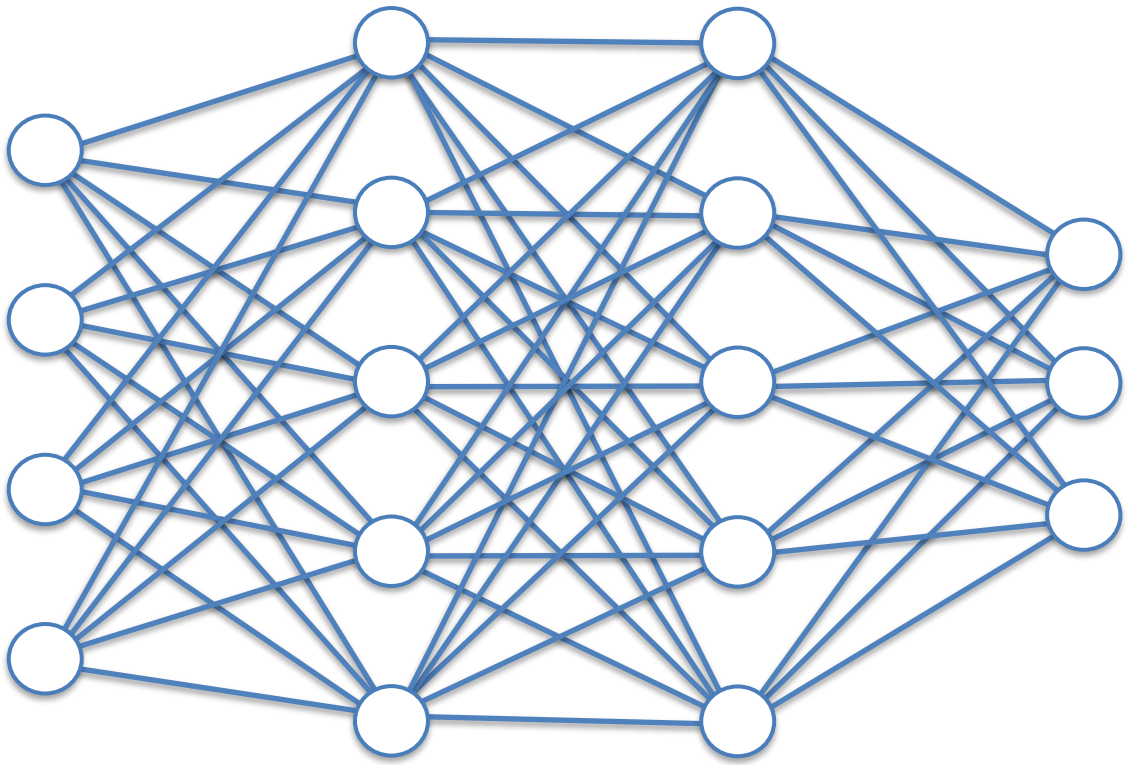
$$y_j = f\left(b_j + \sum_i w_{ij} x_i\right)$$

Example of multilayer network



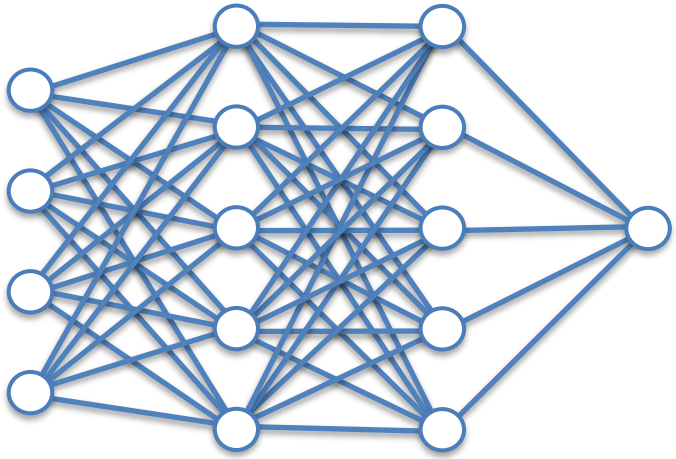
Input layer | Hidden layer | Hidden layer | Output layer

Example of multilayer network



Input layer | Hidden layer | Hidden layer | Output layer

Modification of network weights and impact on output



- Let w_{ij}^n be the weights of the network connections (w_{ij}^n: from cell i of layer $n-1$ to cell j of layer n) and b_j^n the associated bias
- Then

$$\Delta_{output} = \sum_{i,j,n} \frac{\partial output}{\partial w_{ij}^n} . \Delta_{w_{ij}^n} + \sum_{j,n} \frac{\partial output}{\partial b_j^n} . \Delta_{b_j^n}$$

Optimization of network parameters

- Training data ⇔ set of examples $\{(x_k, y_k)\}$

  where x_k is a feature vector (i.e., values given to input layer)
  and y_k is the associated output vector (i.e. output values expected for the input x_k).

- Let $F_\theta(x_k)$ be the output vector for the input x_k, and θ be the network parameters

- The prediction error is

$$Err = \sum_k \|F_\theta(x_k) - y_k\|^2$$

- Training the network parameters consists in optimizing the training criterion (which could be the prediction error)

Optimization of network parameters stochastic gradient descent

- Let θ be the vector corresponding to the network parameters w_{ij}^n et b_j^n
- For each example (x_k, y_k)

$$\theta \leftarrow \theta - \eta . \nabla_\theta Err$$

- That is

$$w_{ij}^n \leftarrow w_{ij}^n - \eta . \frac{\partial \| F_\theta(x_k) - y_k \|^2}{\partial w_{ij}^n}$$

- Back propagation of the output prediction error
  - Compute the derivatives successively on each layer through the derivative chain rule
  - For example, if y_j^n is a cell output of layer n, this gives

$$\frac{\partial Err}{\partial w_{ij}^n} = \frac{\partial Err}{\partial y_j^n} . \frac{\partial y_j^n}{\partial w_{ij}^n}$$

- In practice, gradient vectors are cumulated over « mini batches », i.e., on subsets of examples (few hundred of gradient vectors)

Optimization of network parameters examples of criteria

The output prediction error can be

- Squared error

$$Err = \sum_k \|F_\theta(x_k) - y_k\|^2$$

- Cross-entropy

$$Err = -\sum_k [y_k \log F_\theta(x_k) + (1 - y_k) \log(1 - F_\theta(x_k))]$$

Typically used for classifiers where output values are 0 or 1

After training, the network outputs correspond to *a posteriori* probabilities of the classes $\mathrm{P}(c_k|input)$

Many other criteria can be used, depending on the optimization problem

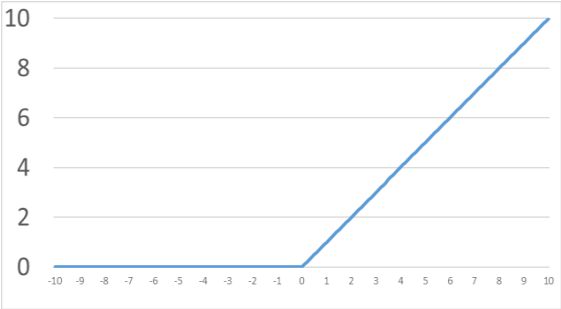
Speech
Basics of speech recognition
Automatic speech recognition
Deep neural networks
Neural networks
Deep, recurrent & convolutional networks
Application to acoustic modeling
Application to language modeling
Extracting other information

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Deep, recurrent & convolutional networks

DNN: Deep Neural Networks

- Typically 4 ou 5 hidden layers, or more
- Generally, ReLU non linear functions on the hidden layers
  - Better behavior of the gradient for training



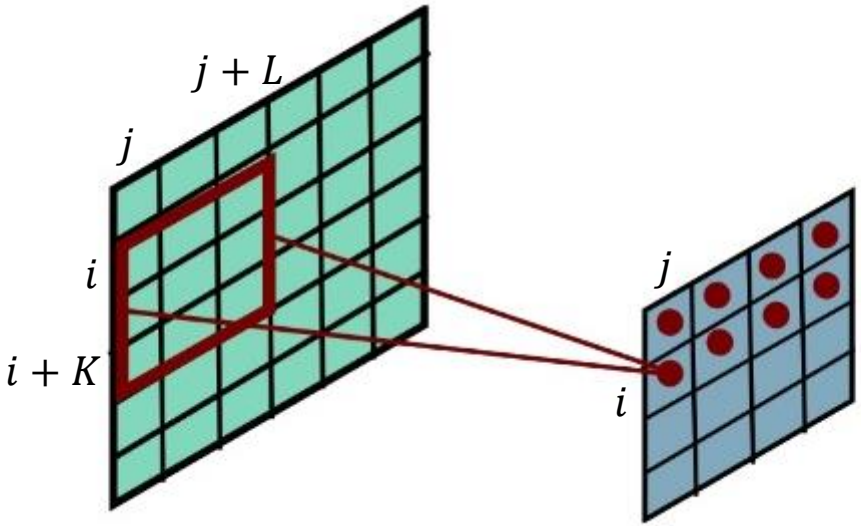
- Usage of GPU (Graphics Processing Units)
  - Notably speed up computations during training

Various types of neural networks

- Towards structures that are more complex than the basic multilayer network
- Combination of standard dense layers (cf. previous slides) with more complex structures such as
  - Convolutional networks
  - Recurrent networks
  - LSTM (Long Short Term Memory) networks

Convolutional neural network

- Each cell of a convolutional layer is connected to a subset of cells of the previous layer

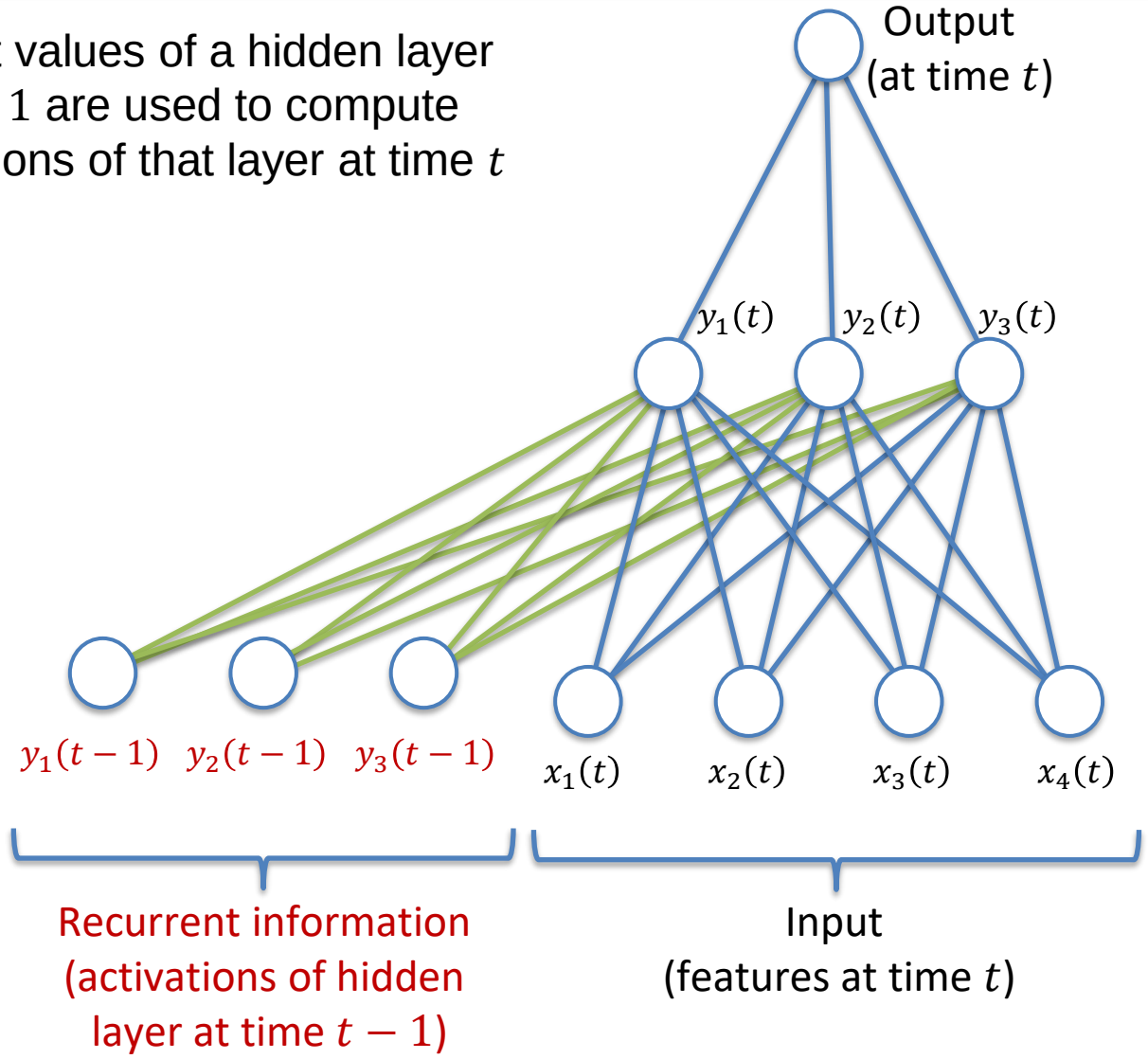


$$y_{i,j} = f\left(\sum_{k=0}^{K}\sum_{l=0}^{L} w_{k,l}.x_{i+k,j+l}\right)$$

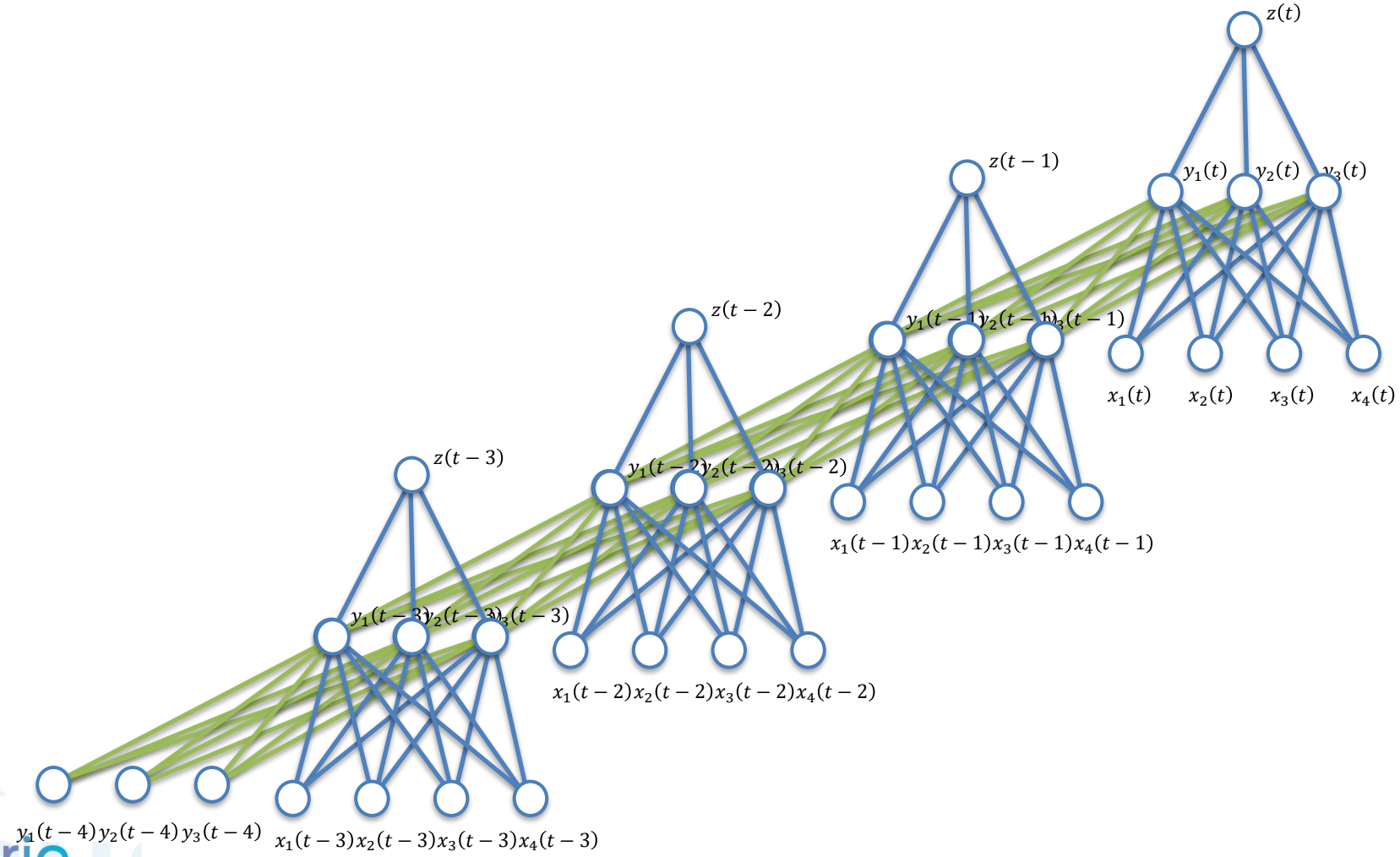
- The same set of weights $\{w_{k,l}; k = 0..K, l = 0..L\}$ is used for computing the activations of all the cells of the convolutional layer

Recurrent neural network

- The output values of a hidden layer at time $t-1$ are used to compute the activations of that layer at time t

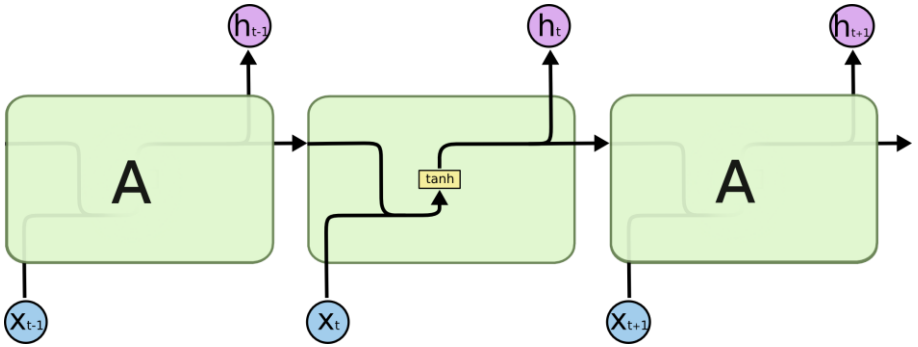


Recurrent neural network

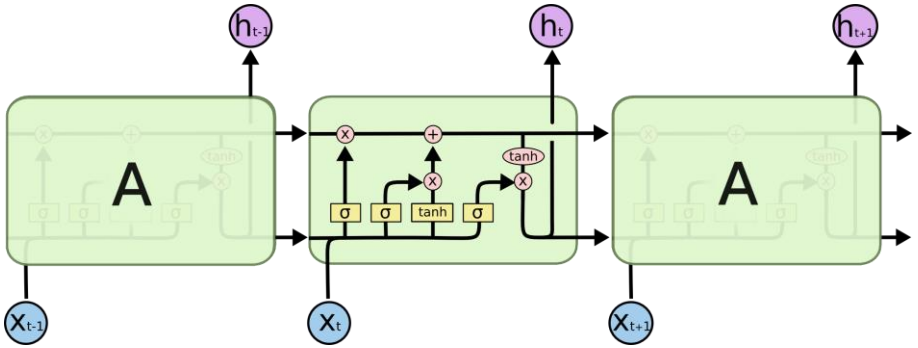


LSTM (Long Short Term Memory) network

- Recurrent network

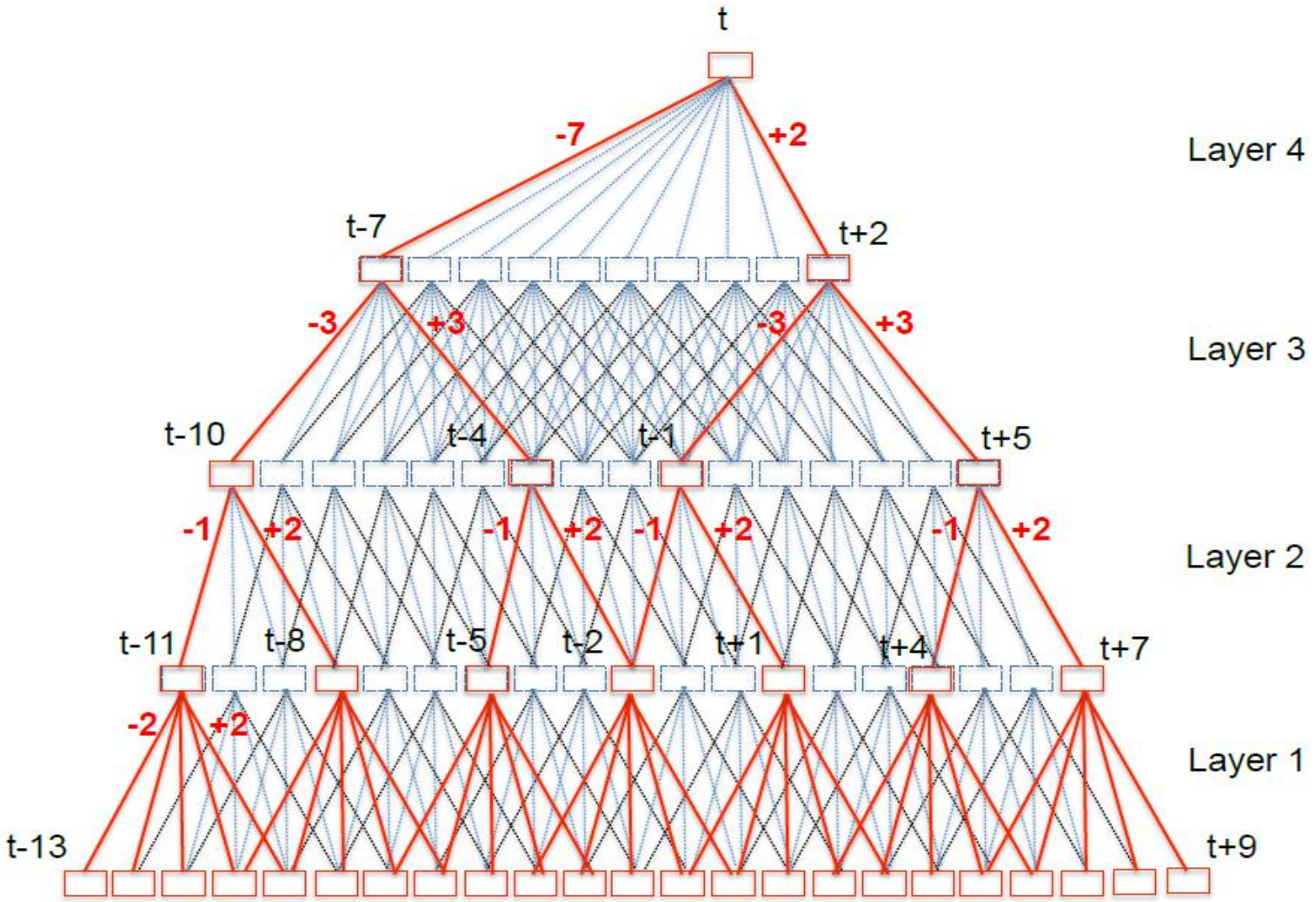


- Complex structure including
  - « *forget gate* » which define how much recurrent information (from past frame) should be kept
  - « *input gate* » which define the new contribution (from current time frame)
  - « *output gate* » which define the output contribution of this cell



- See http://colah.github.io/posts/2015-08-Understanding-LSTMs/ for more details

TDNN – Time Delay Neural Network



Speech
Basics of speech recognition
Automatic speech recognition
Deep neural networks
- Neural networks
- Deep, recurrent & convolutional networks
- **Application to acoustic modeling**
- Application to language modeling

Extracting other information

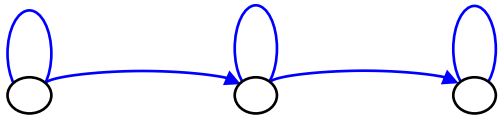
Laboratoire lorrain de recherche en informatique et ses applications

Application to acoustic modeling

Reminder for HMM-GMM decoding

HMM: Hidden Markov Model
GMM: Gaussian Mixture Model
⇔ densities

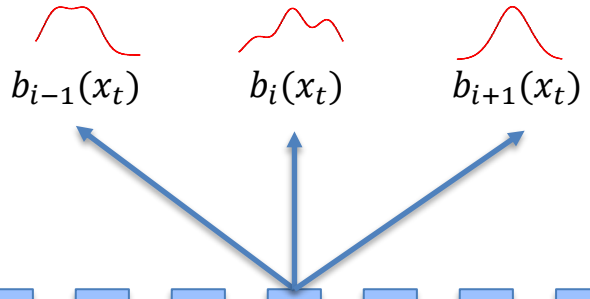
Graph



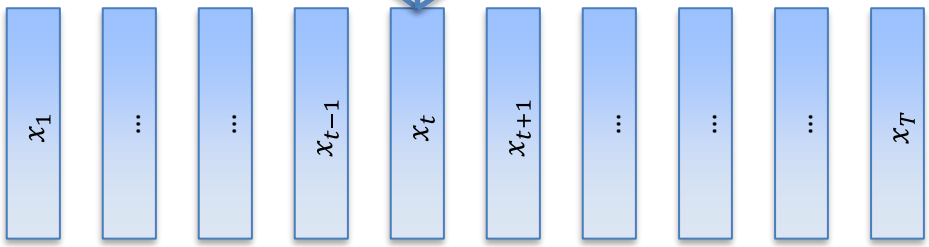
$p(x_t|q_{i-1}) = b_{i-1}(x_t)$ $p(x_t|q_i) = b_i(x_t)$ $p(x_t|q_{i+1}) = b_{i+1}(x_t)$

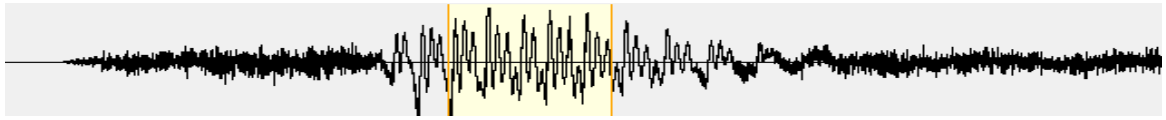
Decoding:
Update of scores $\delta_t(j)$
(*algo. Viterbi*)

Densities (pdfs)



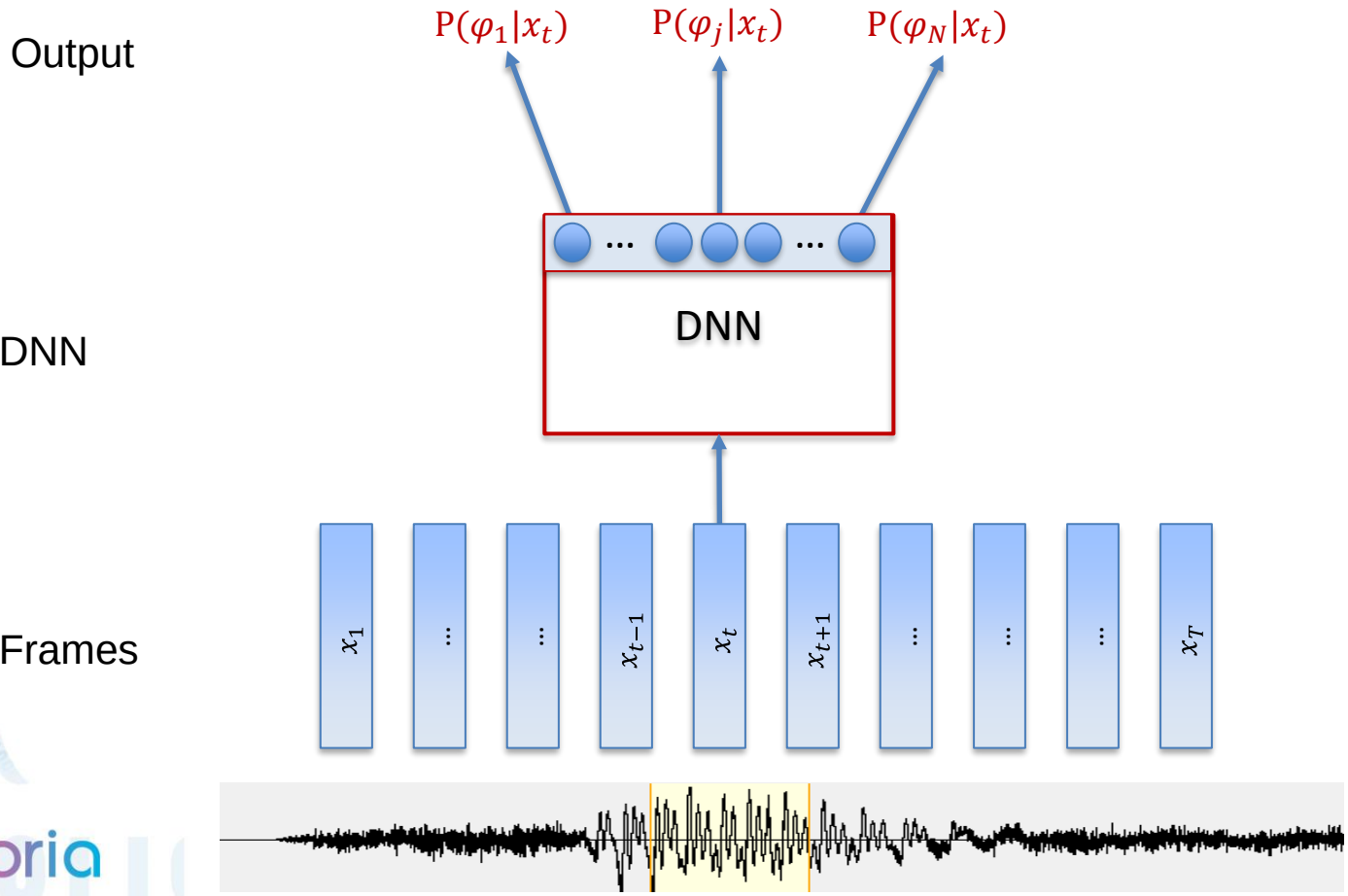
Frames





DNN – with output associated to phonemes

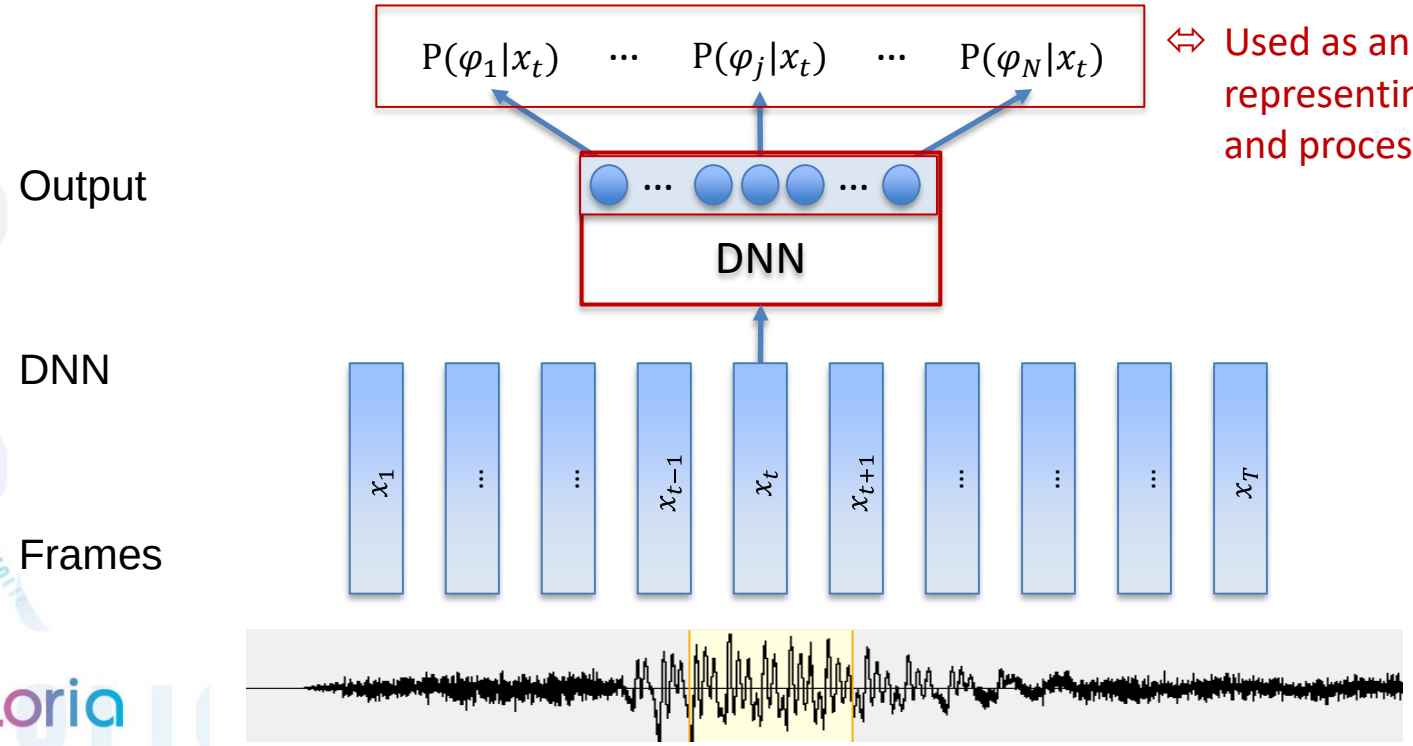
For a DNN trained with the cross-entropy criterion, the output of the neural networks provides *a posteriori* probabilities of the classes (here, phonemes)



Tandem approach

In the Tandem approach, the output of the neural network (i.e. phone probabilities) is used as a vector of features.

These feature vectors are then processed by an HMM



⇔ Used as an input vector representing the frame x_t, and processed by an HMM

Hybrid HMM-DNN model

HMM: Hidden Markov Model
DNN: Deep Neural Network
⇔ replace the GMM densities

The DNN replaces the GMM densities, but

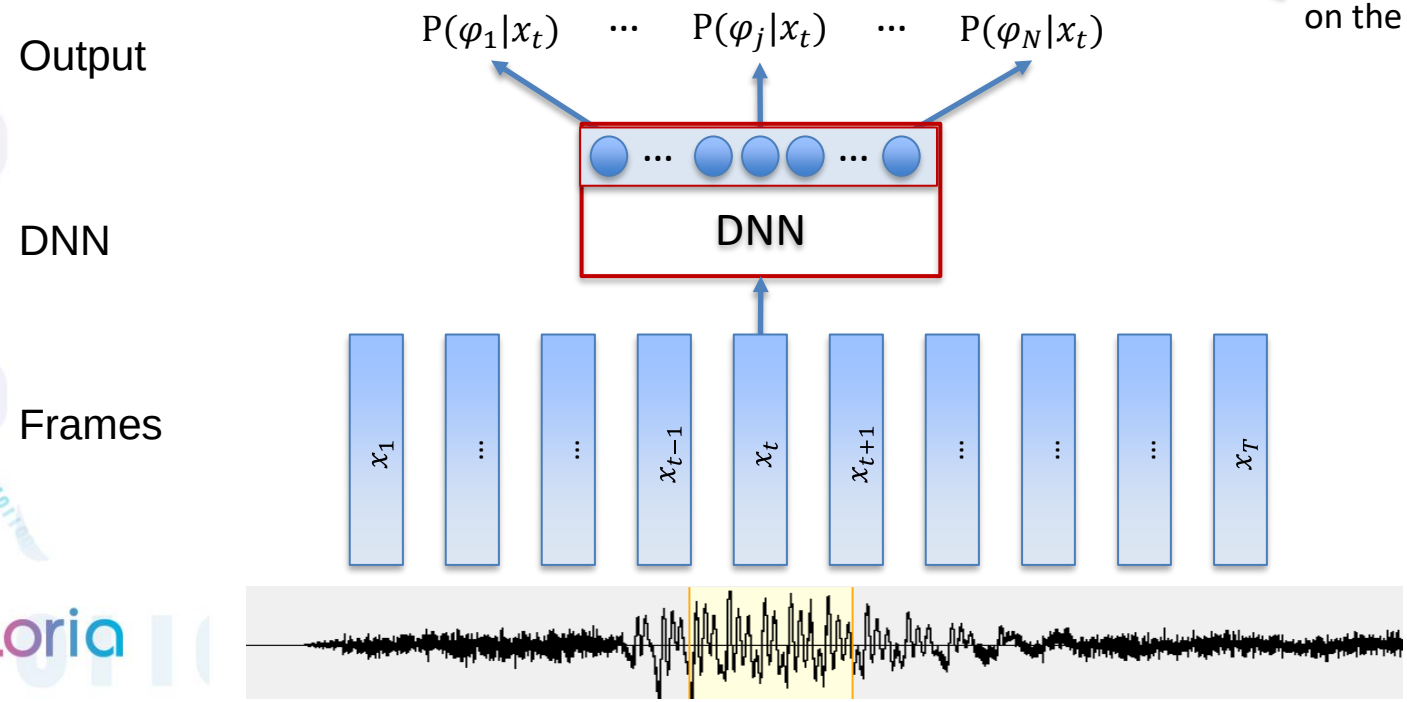
- GMM densities give $b_j(x_t) \Leftrightarrow \mathrm{p}(x_t|q_j)$
- DNN, trained with cross entropy, gives $\mathrm{P}(\varphi_j|x_t)$
- Solution (Bayes rule)

$$\mathrm{p}(x_t|q_j) = \mathrm{p}(x_t|\varphi_j) = \frac{\mathrm{P}(\varphi_j|x_t).\mathrm{P}(x_t)}{\mathrm{P}(\varphi_j)} \propto \frac{\mathrm{P}(\varphi_j|x_t)}{\mathrm{P}(\varphi_j)}$$

Does not depend on the states
➔ can be ignored

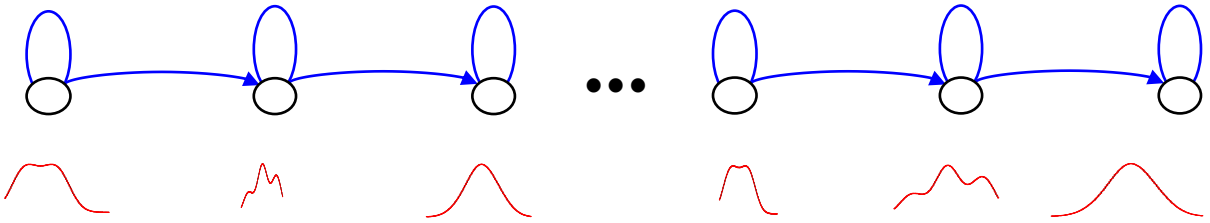
DNN output

Estimated from statistics on the training data

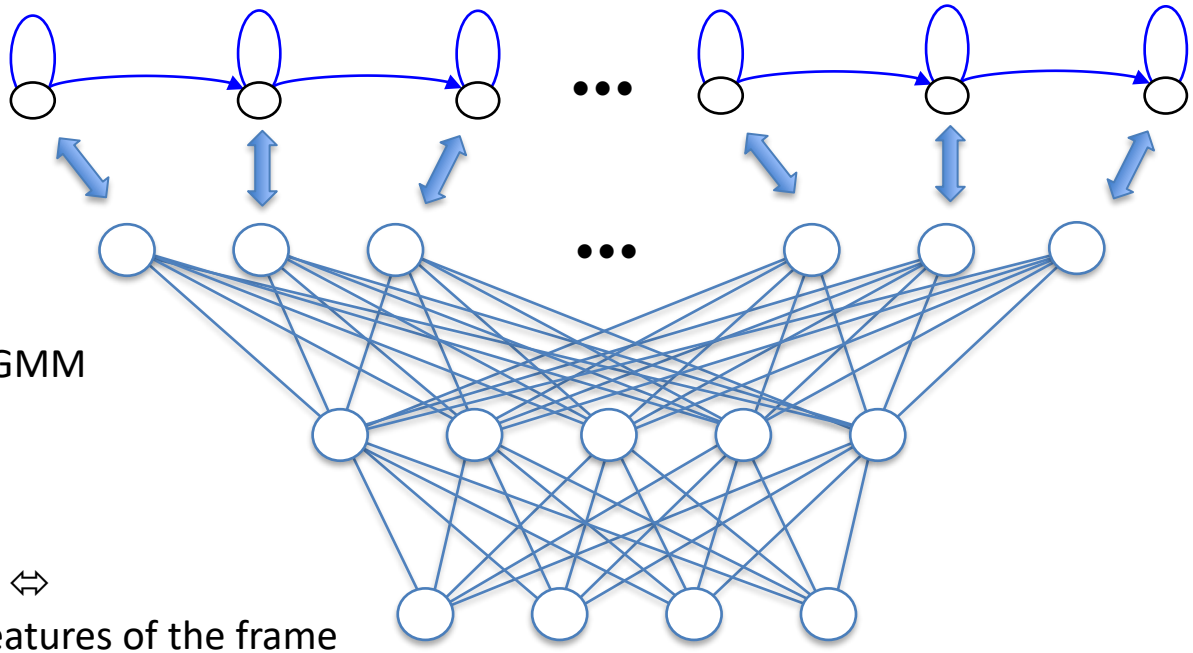


Hybrid HMM-DNN model

- Gaussian mixture model densities



- are replaced by DNN outputs



Ouput of a DNN
Each output cell of the DNN corresponds to a share density of the initial HMM-GMM

Input cells ⇔
acoustic features of the frame

Hybrid HMM-DNN model

- First build and train a standard HMM-GMM model
- Get the alignment between the acoustic frames and the shared densities
  ➔ this defines the output targets for training the neural network
- Train the neural network…
- Typically (a few years ago)
  - DNN with 5 hidden layers, each layer having 2048 cells
  - And input features corresponding to an 11 frames window (i.e., 5 frames before and 5 frames after the current frame)
- And now variants using TDNN or convolutive networks or …

Speech recognition performance

Word error rates	Telephone conversations (anglais)	Telephone conversations (chinois)	SMS dictation (chinois)
HMM – GMM (maximum likelihood training)	35.4%	53.6%	24.4%
HMM – GMM (discriminative training *min. phone error*)	32.8%	48.7%	22.3%
HMM – DNN ***(relative WER reduction)***	23.7% ***28%***	34.8% ***28%***	16.8% ***25%***

Clear reduction of the word error rate (WER reduction of 25% or more) when using DNN for acoustic modeling instead of GMM densities

Human vs machine

- Speech corpus : telephone conversations

Word Error Rates	Switchboard	Call Home
Professional transcribers	5,9%	11,3%
Automatic speech recognition (combination of many NN-based systems, trained on large data sets)	5,8%	11,0%

(2017 – Microsoft)

- The results obtained with a combination of many speech recognition systems get similar to those of professional transcribers

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Application to language modeling

Word embedding

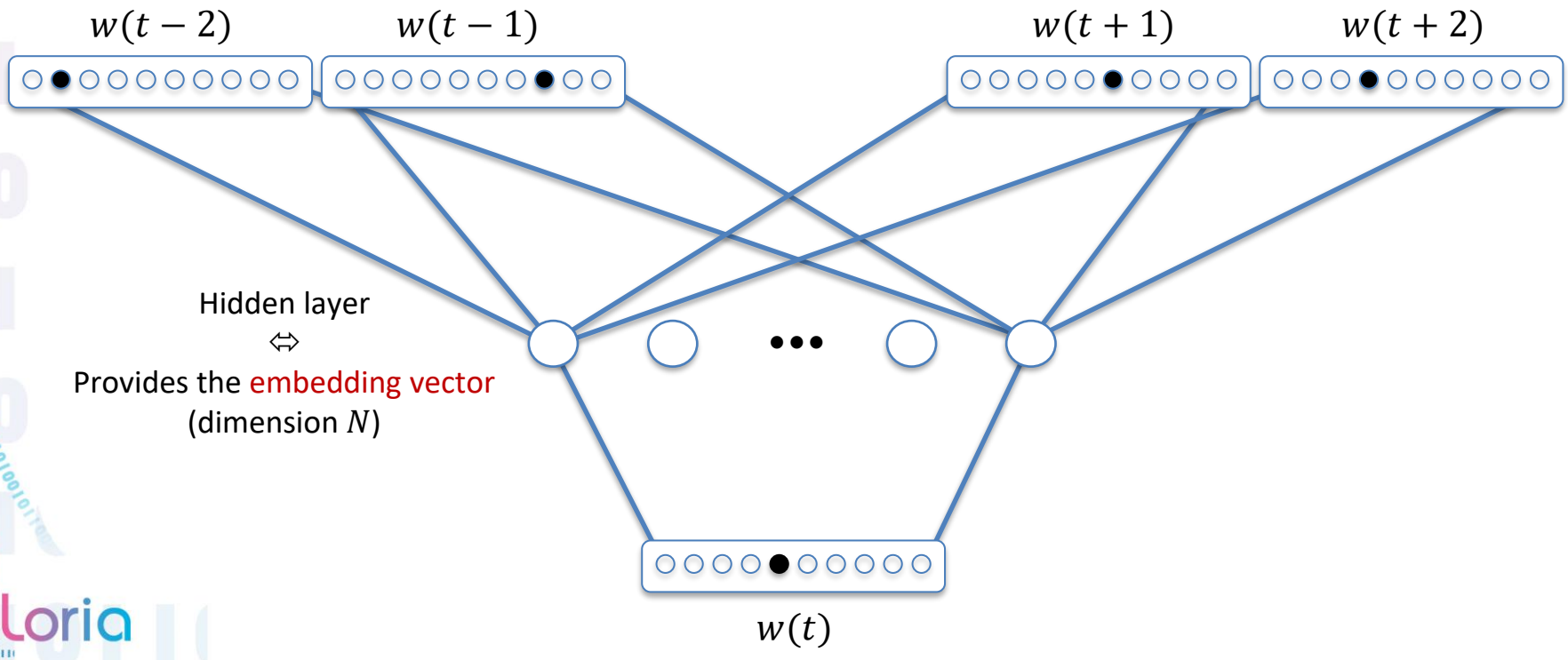
- Projection of word items in a continuous space
  « word » ⇔ vector

- Two approaches relying on neural networks are frequently used (cf. word2vec):
  - *Skip gram model*
  - *Continuous bag of words model*

- Approaches sometimes also called
  - *distributional semantic model*
  - *semantic vector space*
  - *word space*

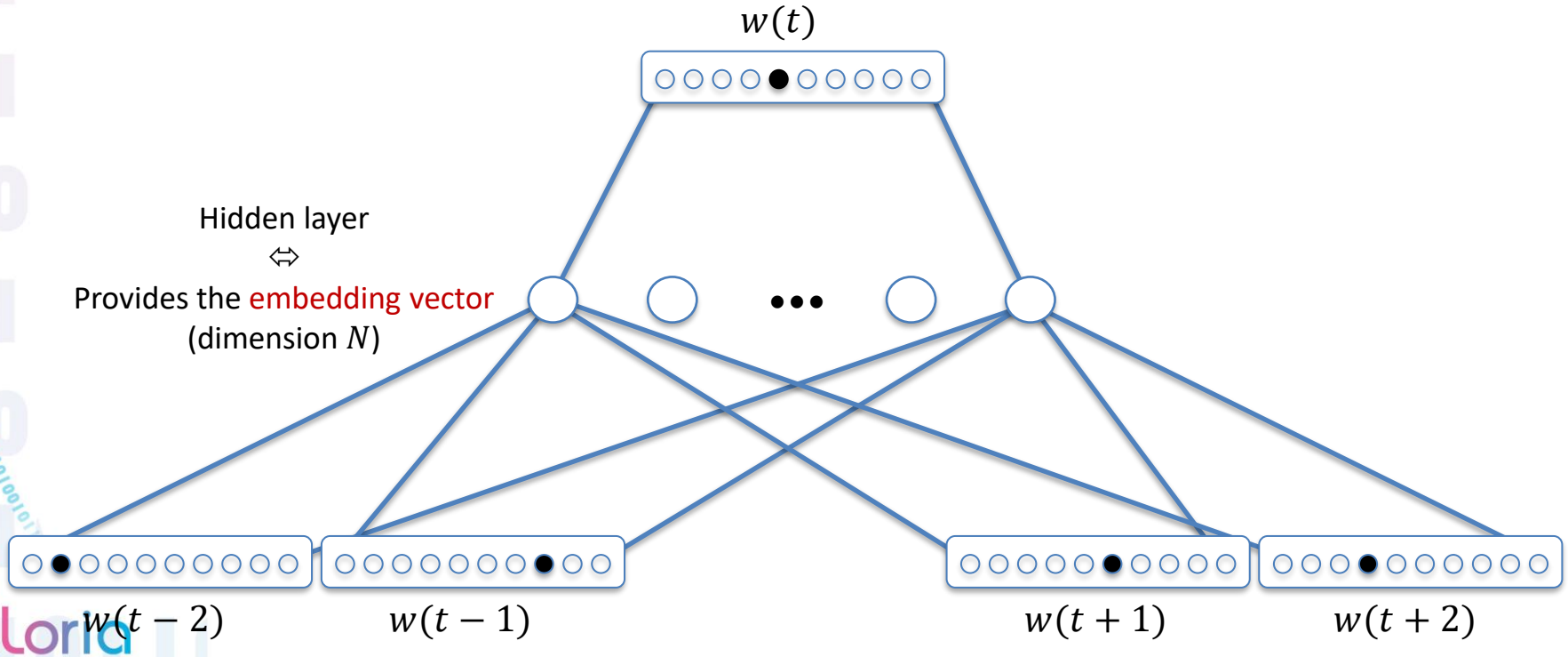
Skip gram model

- Each word (input and output) is represented by a one hot encoding vector (vector of size V, one component equal to 1, all the other set to 0; where V is the vocabulary size)
- Mapping on a hidden layer of arbitrary size N
- Network trained to predict neighbor words from the current one



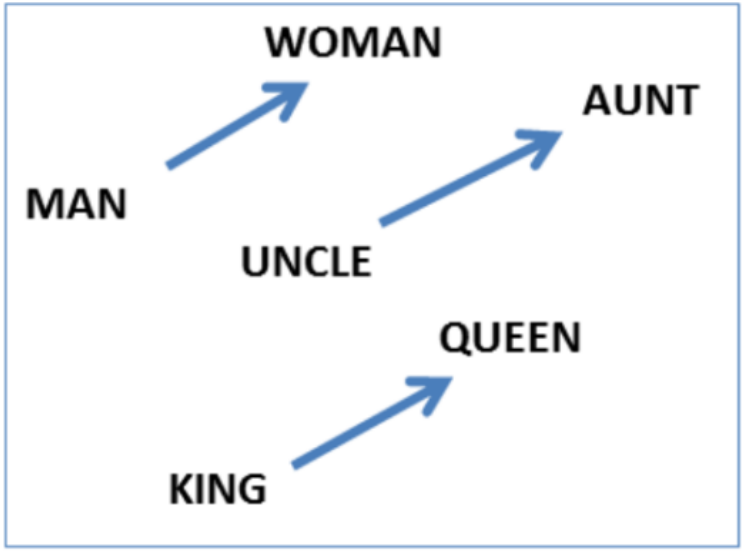
Continuous bag of words model

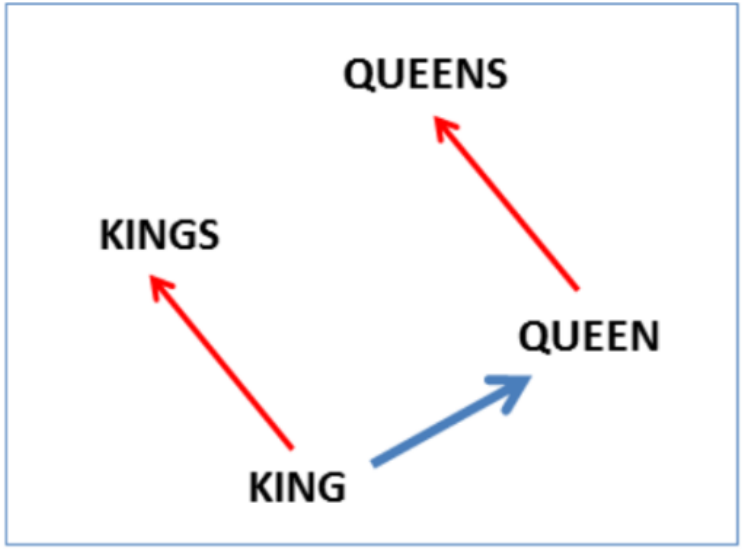
- Each word (input and output) is represented by a one hot encoding vector
- Mapping on a hidden layer of arbitrary size N
- Network trained to predict the current word from neighbor words (weight matrices are shared, sum of neighbor word contributions, so word order ignored, hence the name: *bag of words*)



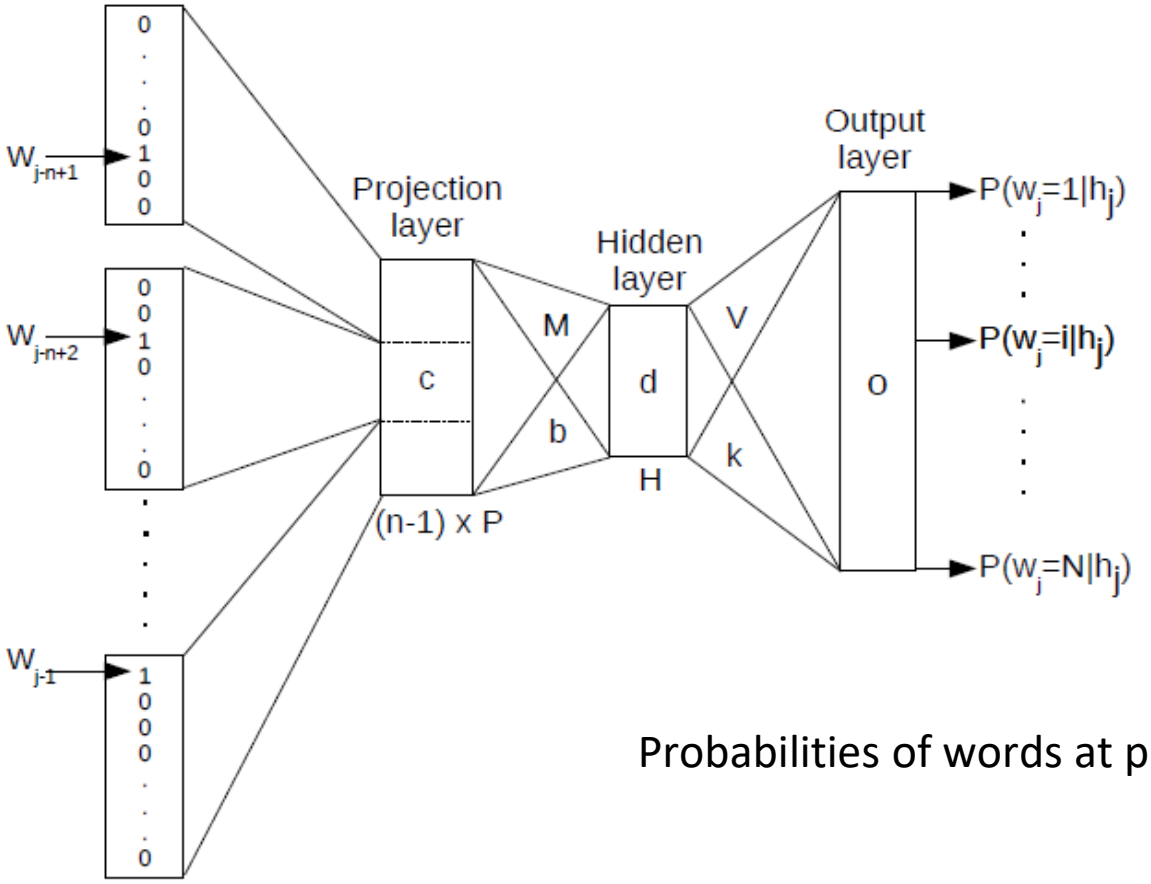
Some properties of « word embedding »

- Vector relations between words
  cf. https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/





Example of neural network language model



Probabilities of words at position j

Previous words $w_{j-n+1}, \ldots, w_{j-1}$
at position $j-n+1, \ldots, j-1$

Remarks

Neural network language model

- Size of the output layer is equal to the number of considered words
- Softmax normalization ($z_j \rightarrow \frac{e^{z_j}}{\sum_k e^{z_k}}$) of output values
- Computation time is significantly reduced when only the most frequent words are considered
- Neural network language model are generally combined with conventional n-gram language models

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Extracting other information

Laboratoire lorrain de recherche en informatique et ses applications

Paralinguistic information

Paralinguistic

- Non verbal information conveyed by speech which also contributes to the communication

- Bring information on
  - Emotional state of the speaker
  - Biological characteristics of the speaker
    - Gender
    - Age
  - Social and cultural characteristics
    - Accent
  - Speaker Health
  - Speaker stress
  - Etc.

Speech and emotions

- Contenu linguistique cohérent avec l'émotion exprimée

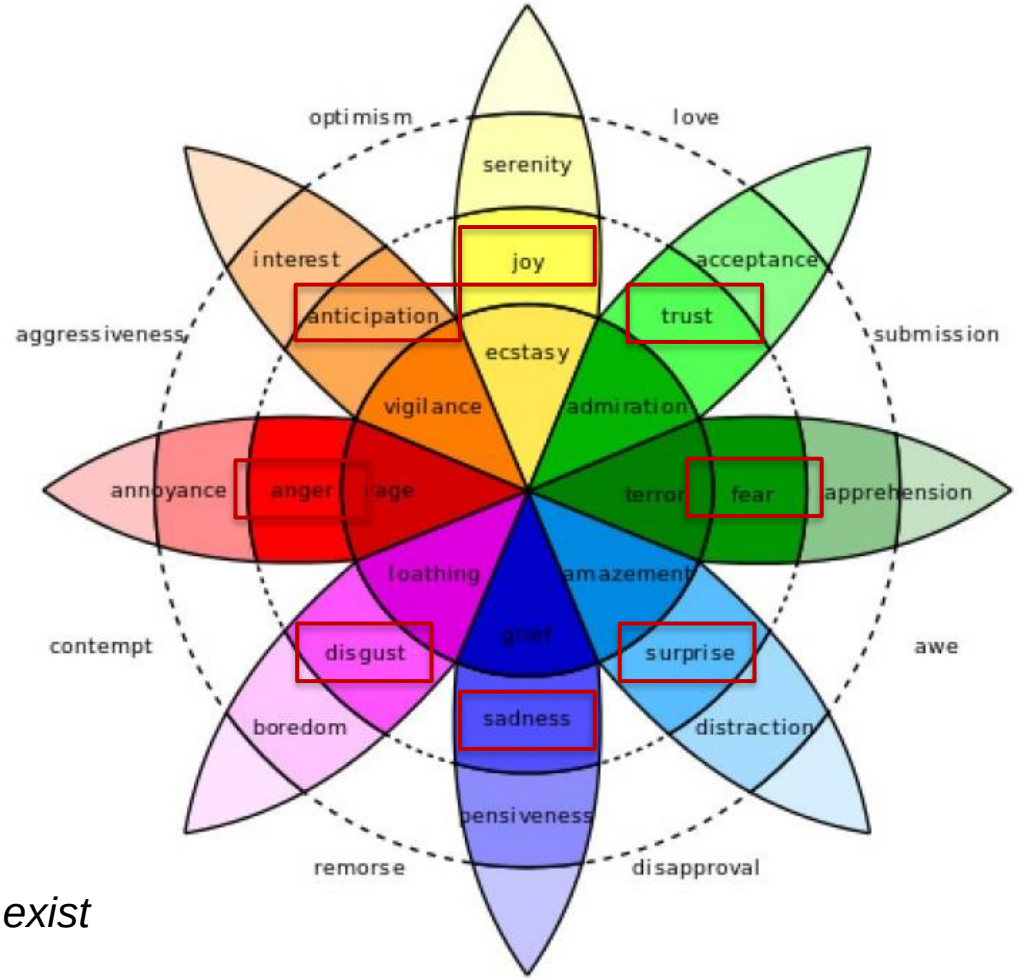
Anger	Fear	Joy	Sadness	Disgust	Surprise
Neutral	Neutral	Neutral	Neutral	Neutral	Neutral

- Contenu linguistique sans lien avec l'émotion exprimée

Anger	Fear	Joy	Sadness	Disgust	Surprise
Neutral	Neutral	Neutral	Neutral	Neutral	Neutral

Robert Plutchik emotion wheel

- Propose eight basic emotions
  - Fear
  - Anger
  - Joy
  - Sadness
  - Trust
  - Disgust
  - Anticipation
  - Surprise
- That
  - Work by pairs
  - Differents intensity degrees
  - Can be combined
- *Remark: other classifications exist*



Expressive speech corpus

Various approach are used to collect expressive speech data

- In natural situation
  - Does not introduce any bias
  - But, speech quality problems (speech recording)
  - And, content is not controled
- Acted speech
  - The speaker read or repeat a text, while expressing the specified emotion
  - Method the most frequently used
  - Sometimes criticized (non natural emotion)
- Trough elicitation
  - Requires a precise scenario that will lead the actor to pronounce the expected sentence while expressing the expected emotion

Recognition of emotions

- Relies on the computation of a large set of features on the considered speech segments (sentences)
  - For example, Interspeech 2010 Paralinguistic Challenge ➔ 1582 measures
    - Energy in frequency bands,
      Cepstral coefficients (MFCC),
      Fundamental frequency (F0),
      ….
    - And for each feature:
      - Mean value, and standard deviation
      - position of maximum and minimum values
      - percentile 1% et 99%
      - Linear regression coefficient
      - …
- Then processing through a classifier (for example, SVM)
- And now, also neural network based approaches

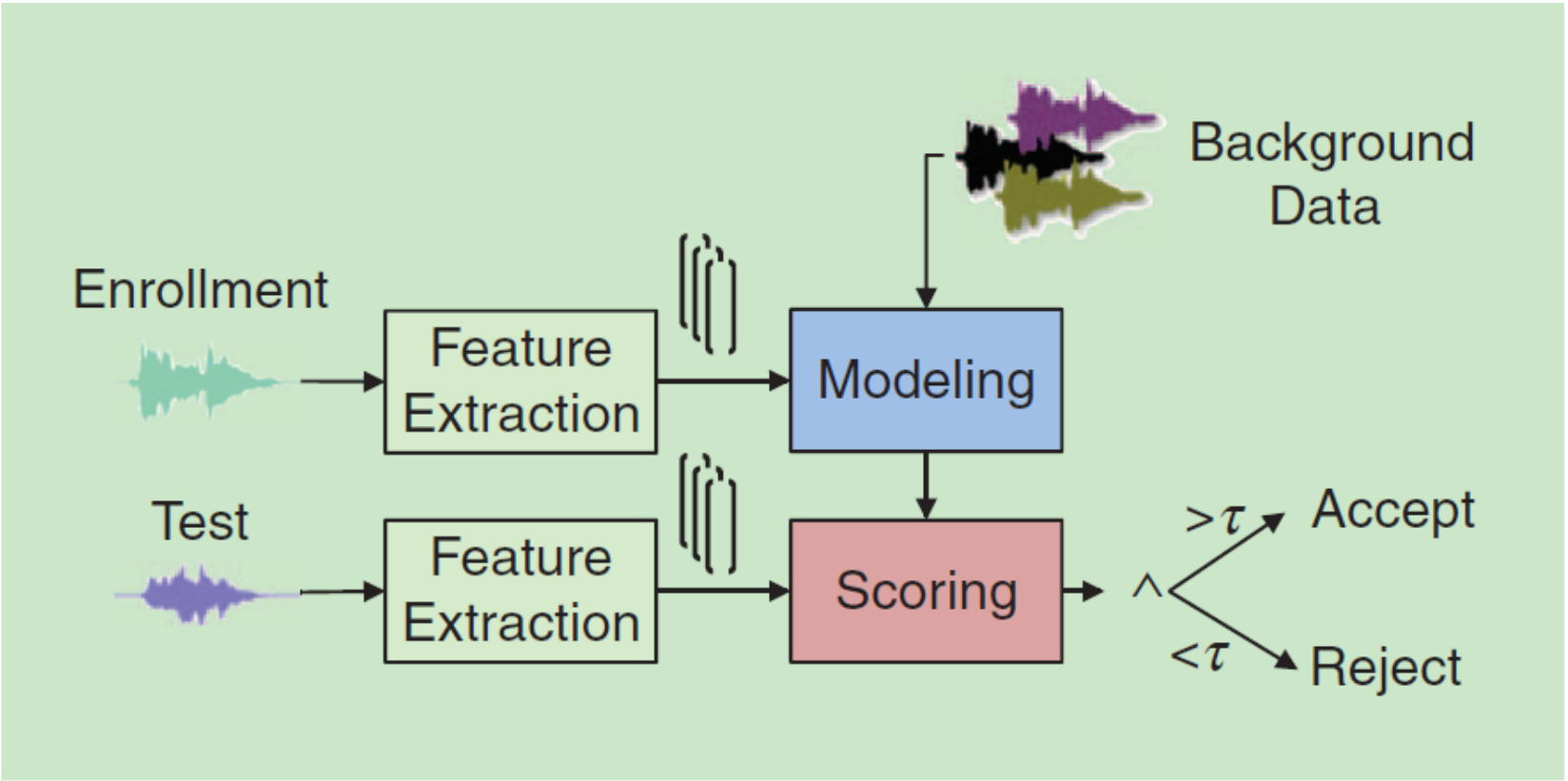
Loria
Laboratoire lorrain de recherche en informatique et ses applications

Speaker and language recognition

Speaker recognition

- Speaker recognition
  - Identify the speaker among a group of known speakers
  - Case of unknown speaker (problem similar to out-of-vocabulary words in ASR)
  - Check that the speaker is really the one he pretends to be
    ⇔ Vocal authentication (or speaker verification)
  - Segmentation in speakers, i.e., identify in an audio document, all the speech segments pronounced by the same speaker (« *diarization* »)
- Operating modes
  - Text-dependent (text known by the system)
  - Text-independent

Speaker verification



Speaker verification

- Speaker representation: was mainly based on i-vectors
  - "World" GMM representing all speakers ➔ m super vector of Gaussian means
  - GMM adaptated on speaker data ➔ μ super vector of adapted Gaussian means
  - i-vector w (dimension 400 to 600 usually) obtained such that

    $$\mu = m + T.w + \epsilon$$

    where T is a projection matrix

- Normalization of scores help improving performance

- Now, neural network based approaches are getting more and more used, also for speaker recognition

Language recognition

- Identify the language spoken
- Phonotactic approach
  - Phonetic speech decoding using one or several phoneme loop models
  - Then computation of the likelihood of the recognized phoneme sequence with respect to phoneme-based language models associated to the various languages
- i-vector based approach
  - Processing similar to what is done for speaker recognition
- And now, neural networks are also getting used for language recogniton

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Conclusions

Continuous speech recognition

- Efficient algorithms and tools for building and optimizing models from data
  - Language ⇔ text corpora
  - Acoustic ⇔ speech corpora (with associated transcription)

- But many choices have to be done by the ASR system developer
  - Type of acoustic features: MFCC, PLP, …. and how many
  - Size of temporal windows (around the current frame)
  - Acoustic model structure
    - Number of states / of densities / of Gaussian components per density
    - Or, type of neural network, number of layers, size of layers

- Some trade-off are necessary
  - Few parameters ⇔ rough modeling but reliable estimation
  - Many parameters ⇔ detailed modeling, but estimation may be unreliable
- Training from speech data leads to good recognition performance on similar speech data (but performance degrade in different/new conditions)

About the approaches

- Hidden Markov model based speech recognition
  - Acoustic coefficients (MFCC, PLP), noise reduction, temporal evolution (derivatives, PCA, LDA, …), normalization (CMN, …)
  - Acoustic modeling: HMM-GMM, subspace GMMs, …
  - Optimization of parameters (training): maximum likelihood, discriminant training
  - Adaptation to speaker or environment: MLLR, MAP, VTLN, SAT, …

- Neural network based speech recognition
  - Replace the Gaussian mixture densities of the approach HMM-GMM
  - Many possible variants: DNN, LSTM, CNN, TDNN, …
  - Training on very large speech corpora
  - Neural network have significantly improved speech recognition performance. ASR performance is getting close to human performance on some tasks

Conclusion

- Machine learning (HMM or DNN) provides an powerful framework for speech recognition
- Training and adaptation procedures are efficient
  - Take benefit of the increase in computer resources
  - But still require adequate speech corpora (corresponding to foreseen usage conditions)
  - Larger and larger speech corpora are used for improving further speech recognition performance
- Speech recognition is getting more and more used, in particular through speech-based assistants

Conclusion

- But performance still degrades in adverse conditions, such as
  - High level noise
  - Hands free distant microphones (reverberation problems)
  - Accents (non native speech)
- Limited vocabulary (even if very large, there is still the problem of person names, location names, …)
- *Still far from a universal recognition system, as powerful as a human listener in all conditions*
- But performance continue to improve…

Laboratoire lorrain de recherche en informatique et ses applications

Annex

Loria
Laboratoire lorrain de recherche en informatique et ses applications

French sounds

Classification of French consonants

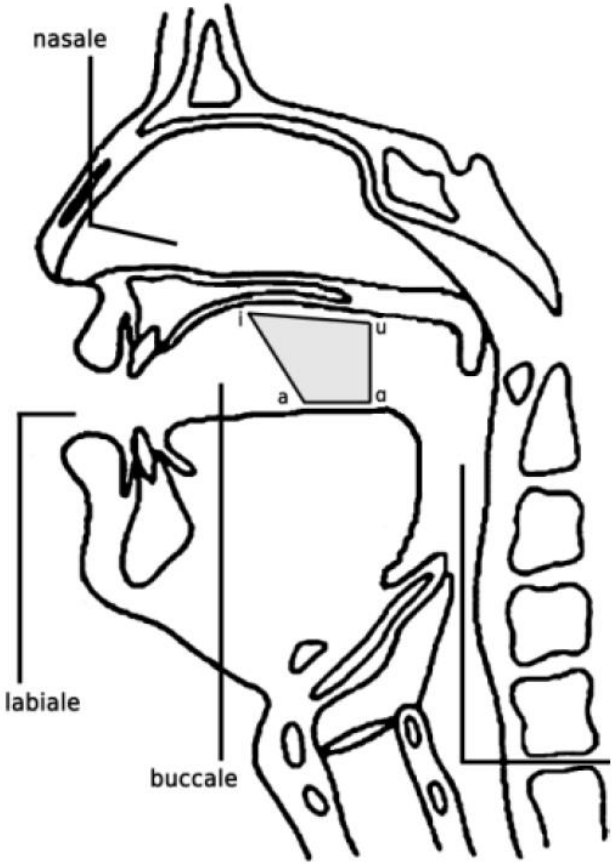
MODE D'ARTICULATION			Bilabiale	Labio-dentale	Dentale	Alvéolaire	Prépalatale	Palatale	Vélaire	Uvulaire			MODE D'ARTICULATION
Occlusive		Médiane	p		t				k		Sourd	Orale	
			b		d				g		sonore		
			m		n			ɲ				Nasale	
Constrictive		Médiane		f		s	ʃ				sourd	Orale	
				v		z	ʒ	j		ʀ	sonore		
		Latérale			l								
		Médiane	ɥ, w					ɥ	w				

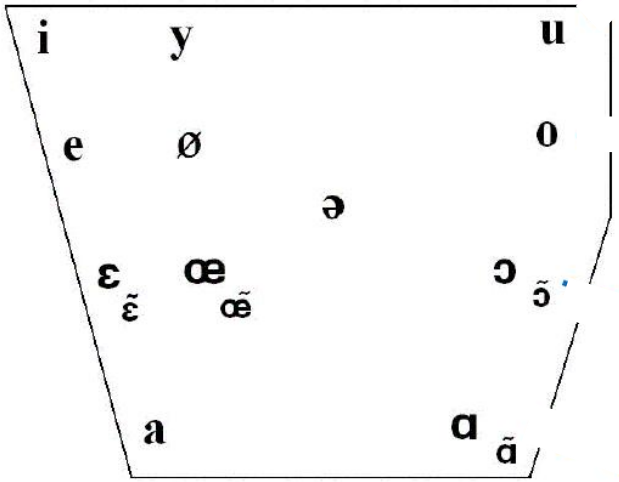
Classification of English consonants

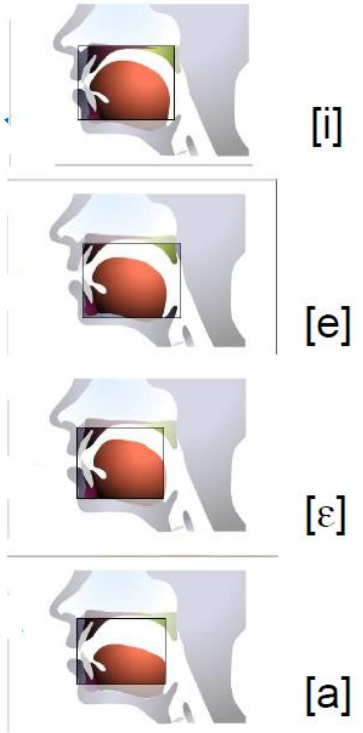
PVM Chart: English

			PLACE							
			LABIAL		CORONAL				DORSAL	
	MANNER	VOICING	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Palatal	Velar	Glottal
OBSTRUENTS	Stop	Voiceless	p			t			k	ʔ
		Voiced	b			d			g	
	Fricative	Voiceless		f	θ	s	ʃ			h
		Voiced		v	ð	z	ʒ			
	Affricate	Voiceless					tʃ			
		Voiced					dʒ			
SONORANTS	Nasal	Voiced	m			n			ŋ	
	LIQUID Lateral	Voiced				l				
	LIQUID Rhotic	Voiced					ɹ			
	Glide	Voiced	w					j	w	

Vowels


nasale
labiale
buccale
pharyngale


i
y
u
e
ø
o
ə
ɛ
œ
ɔ
ɛ̃
œ̃
ɔ̃
a
ɑ
ɑ̃


[i]
[e]
[ɛ]
[a]

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Some applications

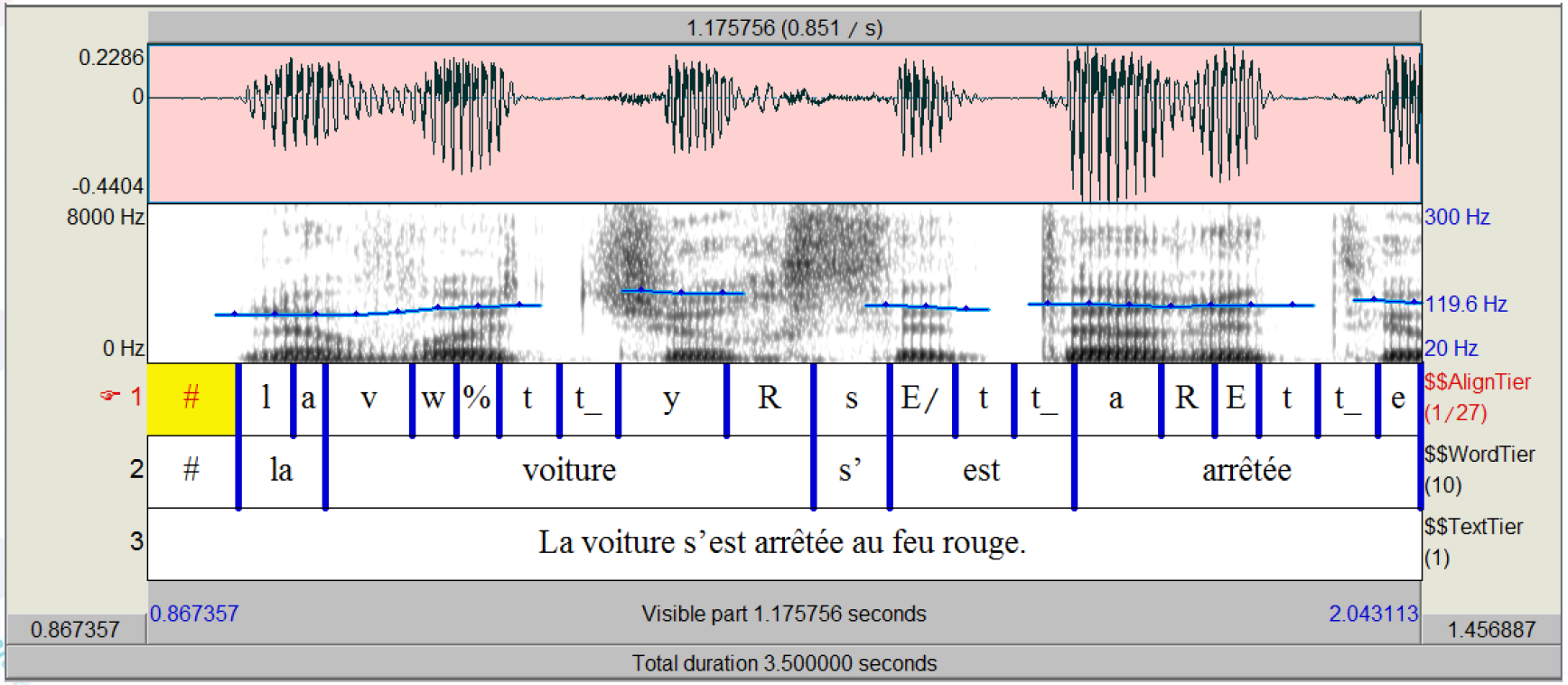
Speech transcription

Typically multi-pass processing

- Diarization: split audio signal in homogeneous segments
  - Detecting speaker changes
  - Identification of signal quality (e.g., studio vs. telephone)
  - Identification of speaker gender (men vs. women)
- Decoding pass
  - Decoding each segment with the most adequate model
- Additional decoding
  - Can take benefit of an unsupervised adaptation of the features or of the model
- Re-scoring
  - Re-scoring a set of recognition hypotheses (n-best or word graph) with more detailed language models
- And possibly combining several systems

Text-speech alignment

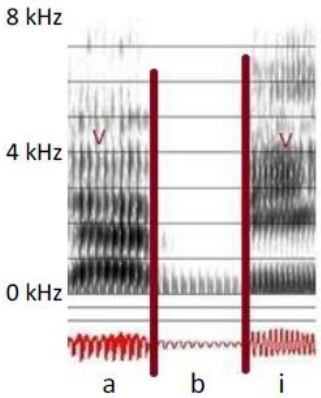
- Aligns a speech signal with the corresponding text
  ➔ provides boundaries of words and phones

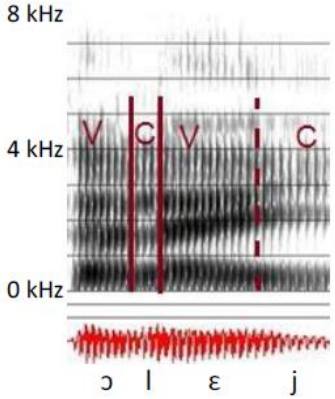


- Applications: speech indexing, linguistic studies, language learning,….

Text-speech alignment & segmentation

- The determination of the boundary between sounds is not always clear, even for a human annotator (e.g., boundary between vowel and semi-vowel)





- The quality of automatic alignments is rather good if
  - Good quality speech signal
  - The text matches the speech content
  - Actual pronunciation matches pronunciation variants present in the lexicon

- Some remaining difficulties: noise, disfluencies, non-native speech, overlapping speech, …

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Miscellaneous

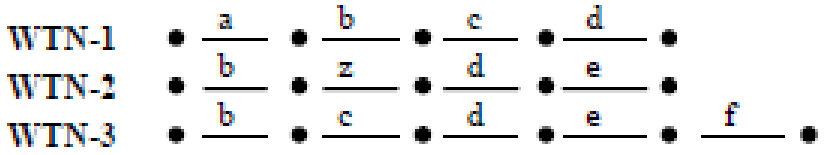
Typical speech recognition output

- Example of French speech recognition output

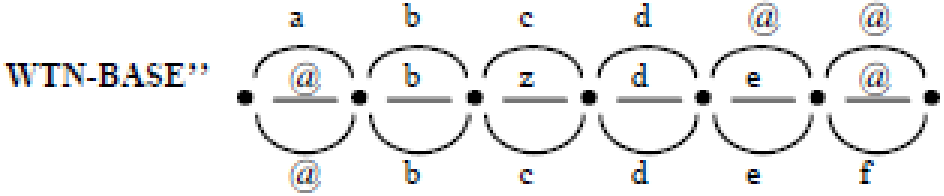
id_of_audio_file	canal	t_start	length	word	confidence
BFMTV_BFMStory_2011-05-31_175900	1	331.460	0.090	le	0.633
BFMTV_BFMStory_2011-05-31_175900	1	331.550	0.320	dernier	0.848
BFMTV_BFMStory_2011-05-31_175900	1	331.870	0.190	point	0.696
BFMTV_BFMStory_2011-05-31_175900	1	332.060	0.570	essentiel	0.798
BFMTV_BFMStory_2011-05-31_175900	1	333.080	0.140	c'est	0.700
BFMTV_BFMStory_2011-05-31_175900	1	333.220	0.070	d'	0.922
BFMTV_BFMStory_2011-05-31_175900	1	333.290	0.370	éviter	0.962
BFMTV_BFMStory_2011-05-31_175900	1	333.730	0.130	un	0.907
BFMTV_BFMStory_2011-05-31_175900	1	333.860	0.550	effondrement	1.000
BFMTV_BFMStory_2011-05-31_175900	1	334.410	0.130	des	0.403
BFMTV_BFMStory_2011-05-31_175900	1	334.540	0.400	cours	0.069
BFMTV_BFMStory_2011-05-31_175900	1	335.250	0.140	des	0.691
BFMTV_BFMStory_2011-05-31_175900	1	335.390	0.490	bovins	0.763
BFMTV_BFMStory_2011-05-31_175900	1	336.210	0.100	en	0.998
BFMTV_BFMStory_2011-05-31_175900	1	336.310	0.410	france	0.998
BFMTV_BFMStory_2011-05-31_175900	1	336.860	0.090	ou	0.554
BFMTV_BFMStory_2011-05-31_175900	1	336.950	0.110	en	0.899
BFMTV_BFMStory_2011-05-31_175900	1	337.060	0.330	europe	1.000

Combining speech recognition outputs

- ROVER: *Recognizer Output Voting Error Reduction*
  - Use a set of ASR outputs



  - Align them



  - And decide what is the best hypothesis on each bloc
  - Criterium: most frequent word, possibly taking into account confidence measures

- Example
  - Using several systems having the following WERs: 31,5%, 30,3%, 29,0%, 29,3%, 31,2%, 30,6% et 30,9%
  - The combination of the outputs leads to WER of 24,5%

Non-supervised training

- Aim to use large speech corpora for which no manual transcription is available
- First, a model is trained using available transcribed speech data
- This model is used to automatically transcribe new speech data
- The most relevant speech data are selected
  - Either through the combination of several decoding approaches
  - Or through the use of confidence measures
- And then used of training or for adapting the model

Laboratoire lorrain de recherche en informatique et ses applications

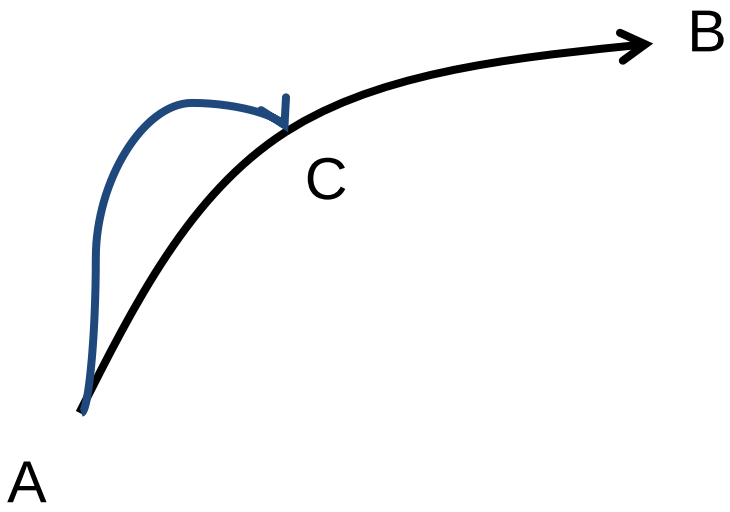
Dynamic programming

Annex – Dynamic programming

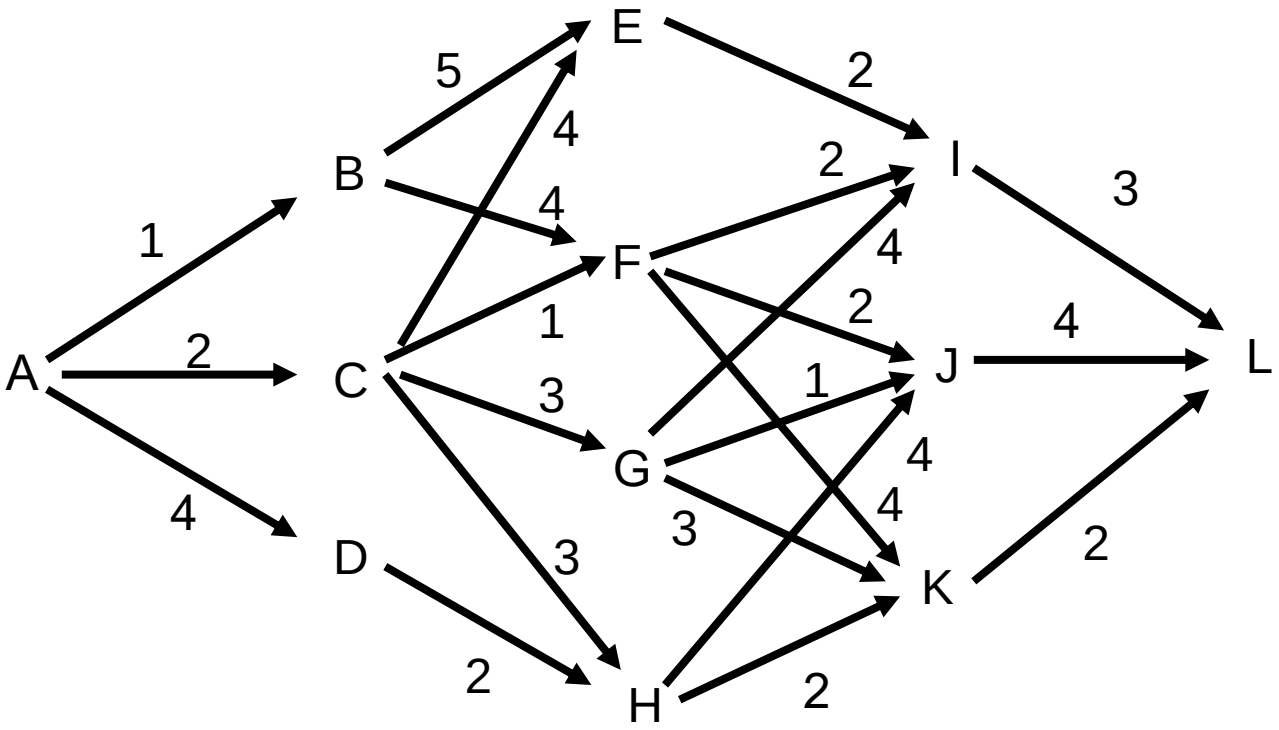
- Best path in a graph
- Comparison / alignment of character strings
- Terminology

Dynamic programming

- Efficient algorithm for finding the best path in a graph
- Rely on the Bellman principle
  ➔ Sub parts of a global optimal path are also optimal

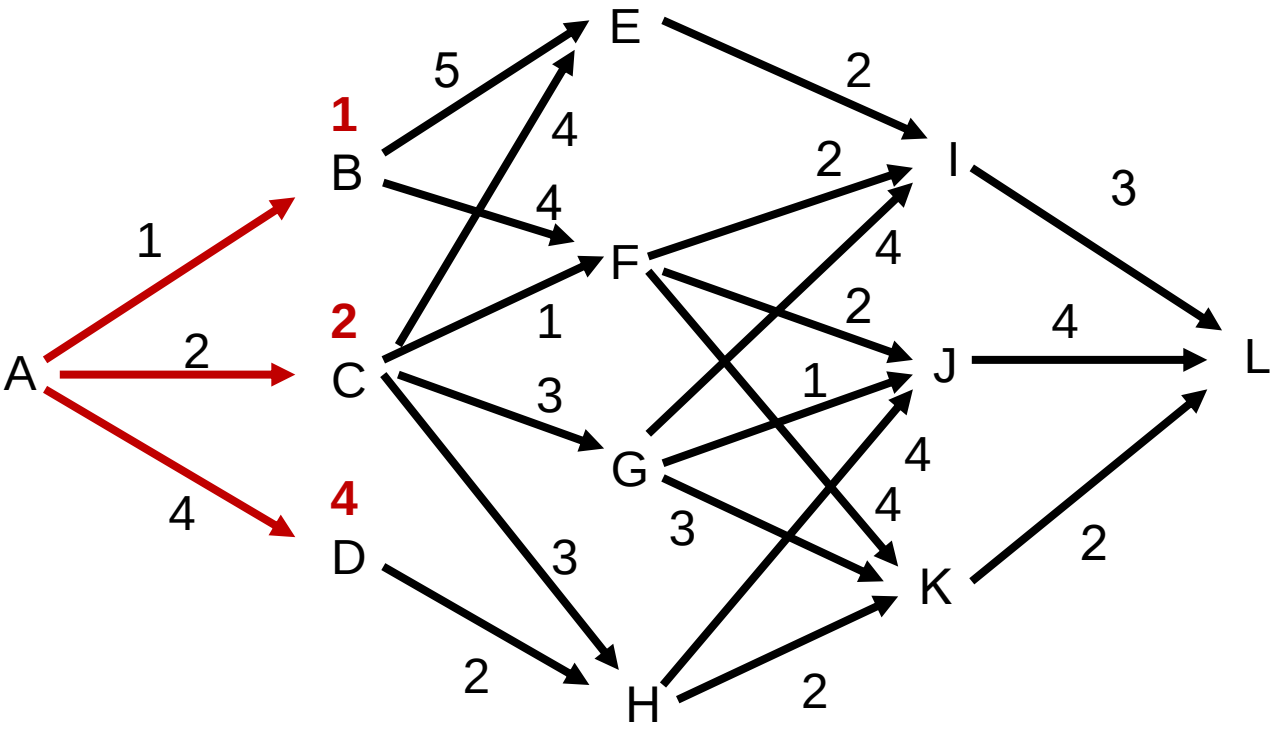


Finding optimal path - example



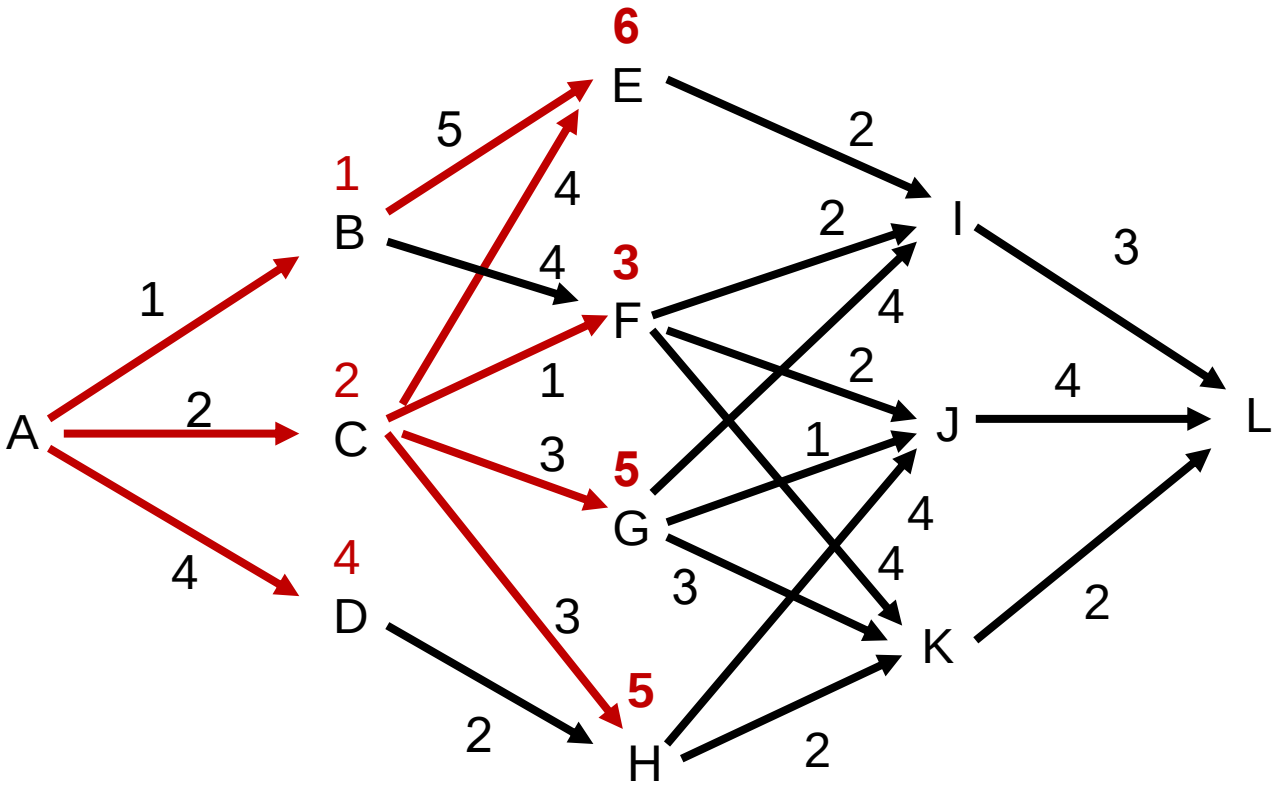
Finding optimal path - example

- Compute iteratively the best path ending on each node



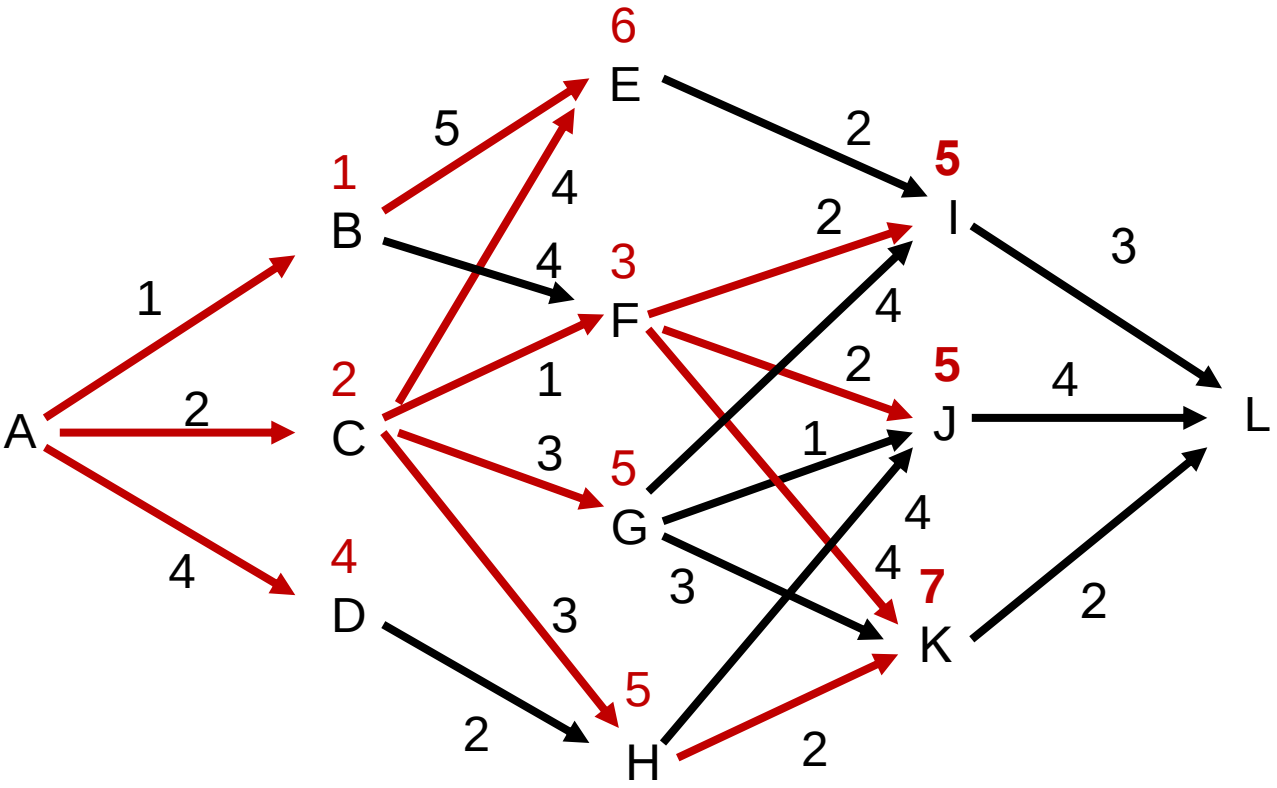
Finding optimal path - example

- … computation continue …



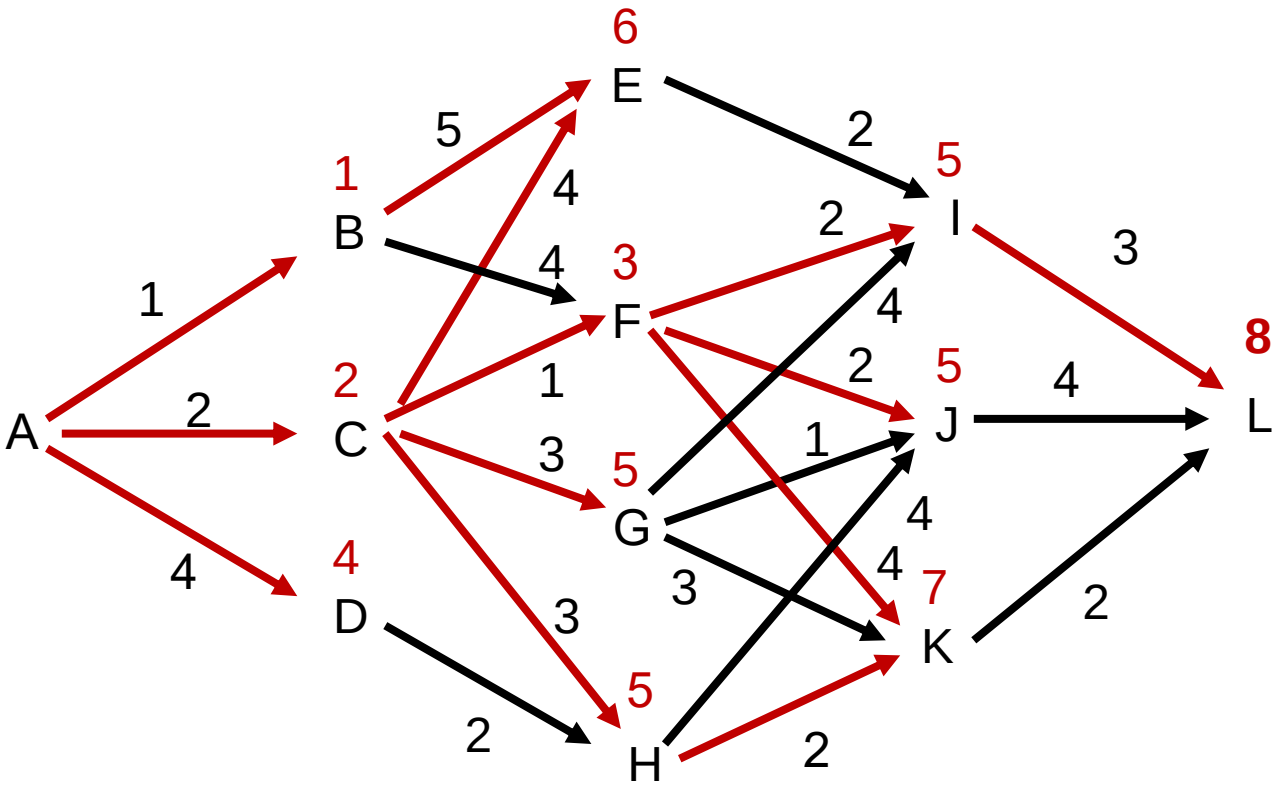
Finding optimal path - example

- … computation continue …



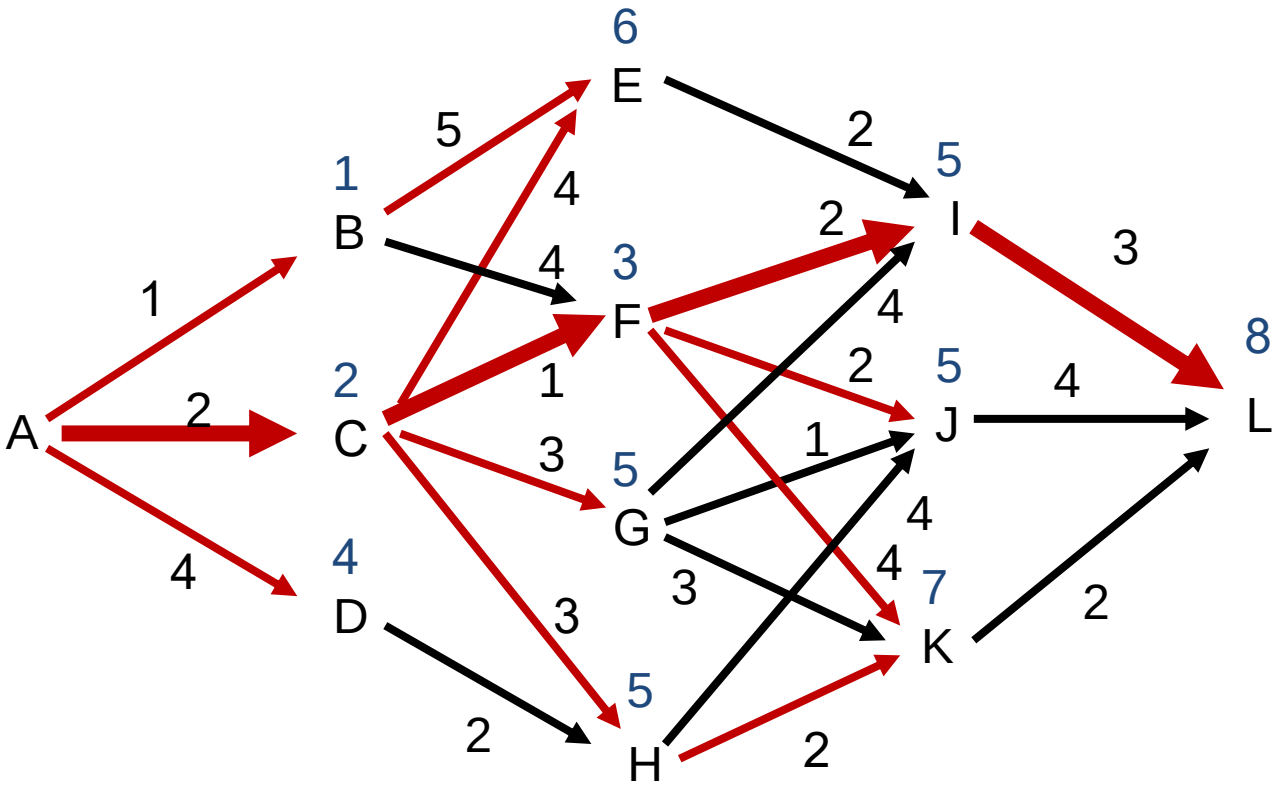
Finding optimal path - example

- … computation continue …



Finding optimal path - example

- Final node reached ➔ The best path can be back-tracked



Comparison of character strings

Example of "distances" between character strings

- *Ref. :* P A R I S
  Test : B A R I F
  => *distance = 2*

- *Ref. :* P A R I S
  Test : P A I F
  => *distance = 2*

- Distortions :
  Deletions *(One letter of the reference missing in test)*
  Insertions *(One extra letter in test)*
  Substitutions *(Correct or Error)*

Dynamic programmic

$$D(I,J) = D(Test,Ref)$$
$$= D(Test[1..I],Ref[1..J])$$

D(I,J) is the « distance » between the I test letters and the J reference letters

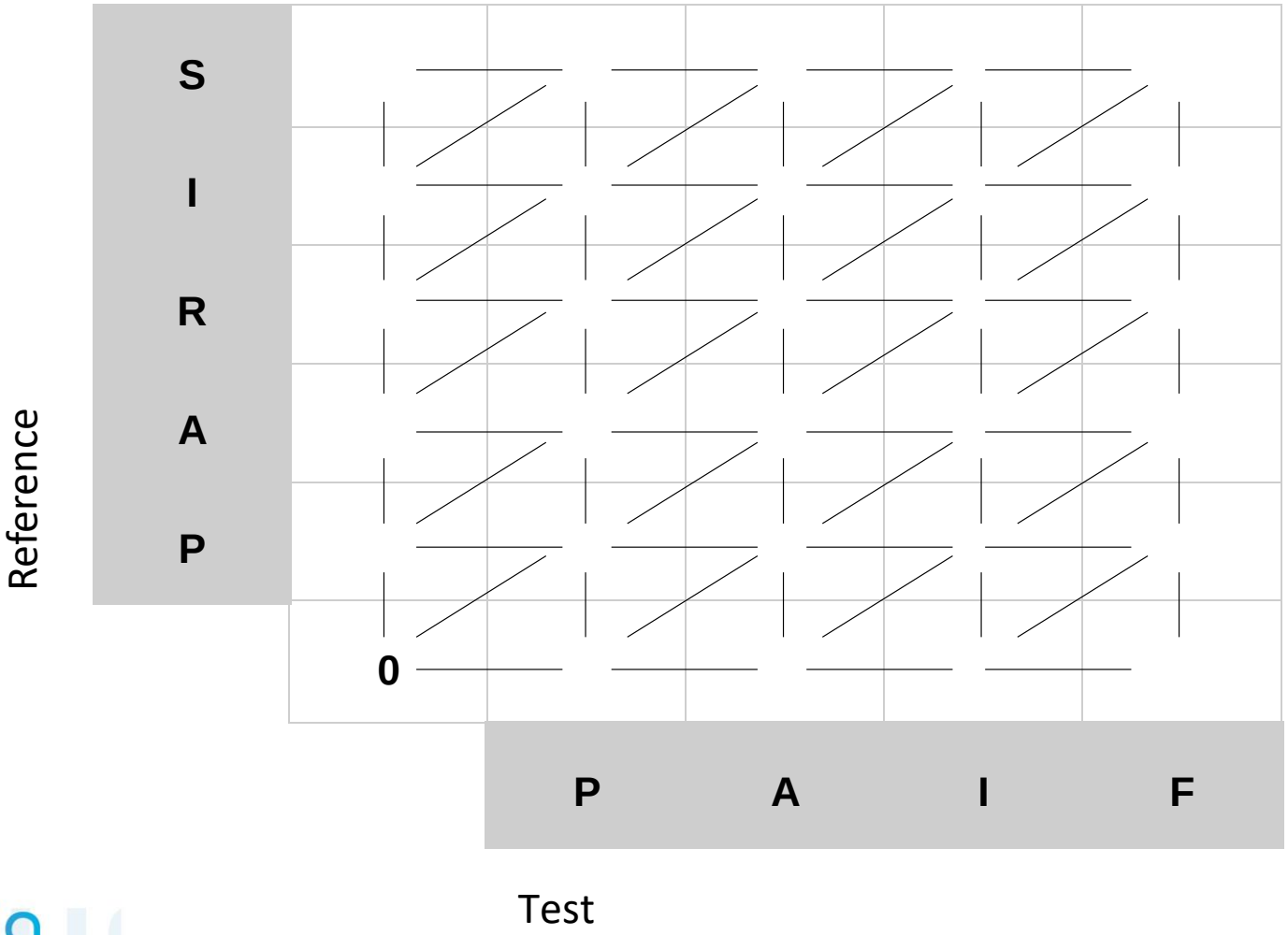
$$D(i,j) = D(Test[1..i],Ref[1..j])$$
$$= \text{Min}\begin{cases} D(Test[1..i-1],Ref[1..j-1]) + d(Test[i],Ref[j]) \\ D(Test[1..i-1],Ref[1..j]) + \textit{CoutInsertion}(Test[i]) \\ D(Test[1..i],Ref[1..j-1]) + \textit{CoutOmission}(Ref[j]) \end{cases}$$

$$d(Test[i],Ref[j]) \Leftrightarrow \textit{CoutSubstitution}.$$

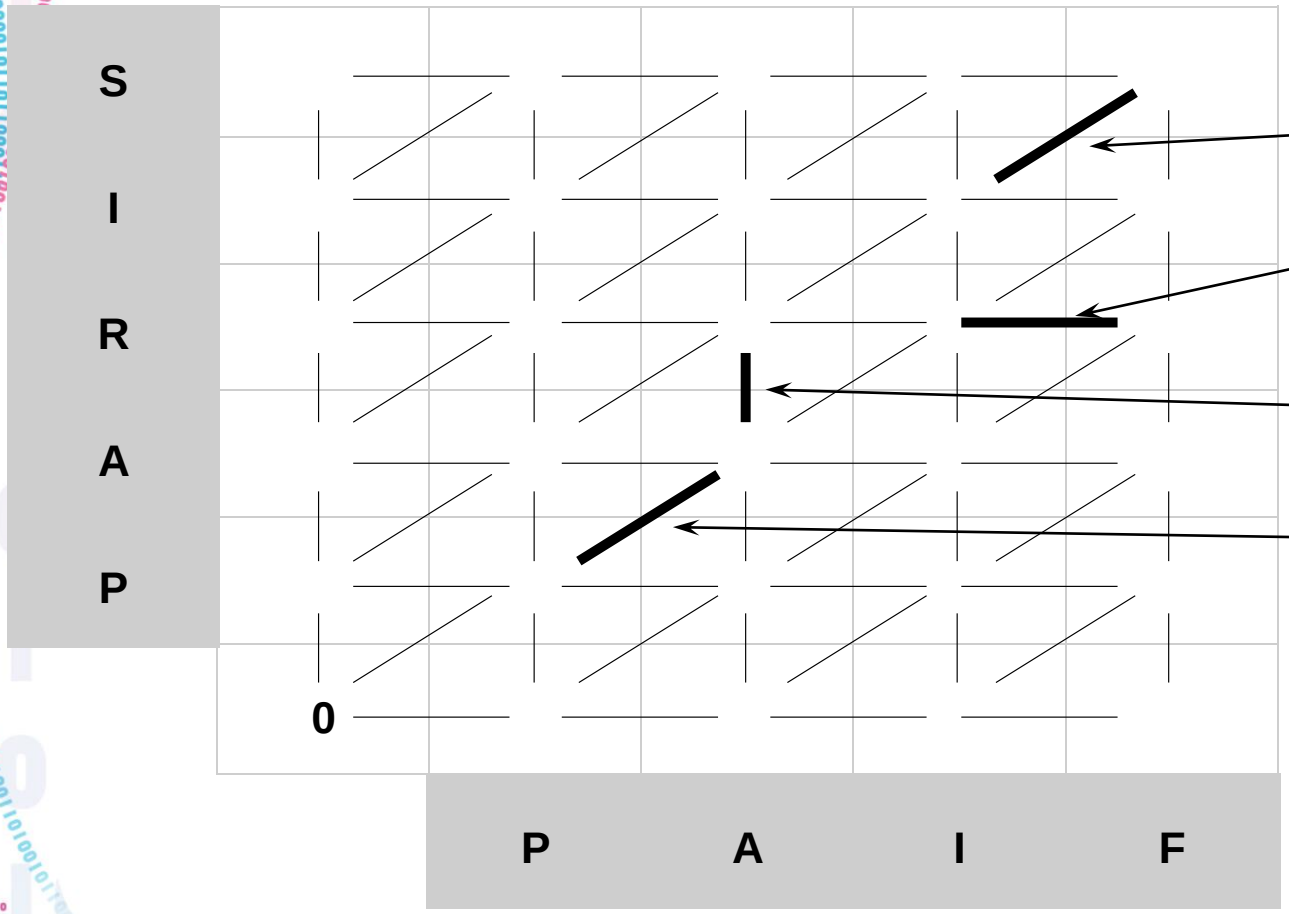
Example of costs:
1 if letters are different
0 if letters are identical

$$D("PAIF","PARIS") = \text{Min}\begin{cases} D("PAI","PARI") + d("F","S") \\ D("PAI","PARIS") + \textit{CoutInsertion}("F") \\ D("PAIF","PARI") + \textit{CoutOmission}("S") \end{cases}$$

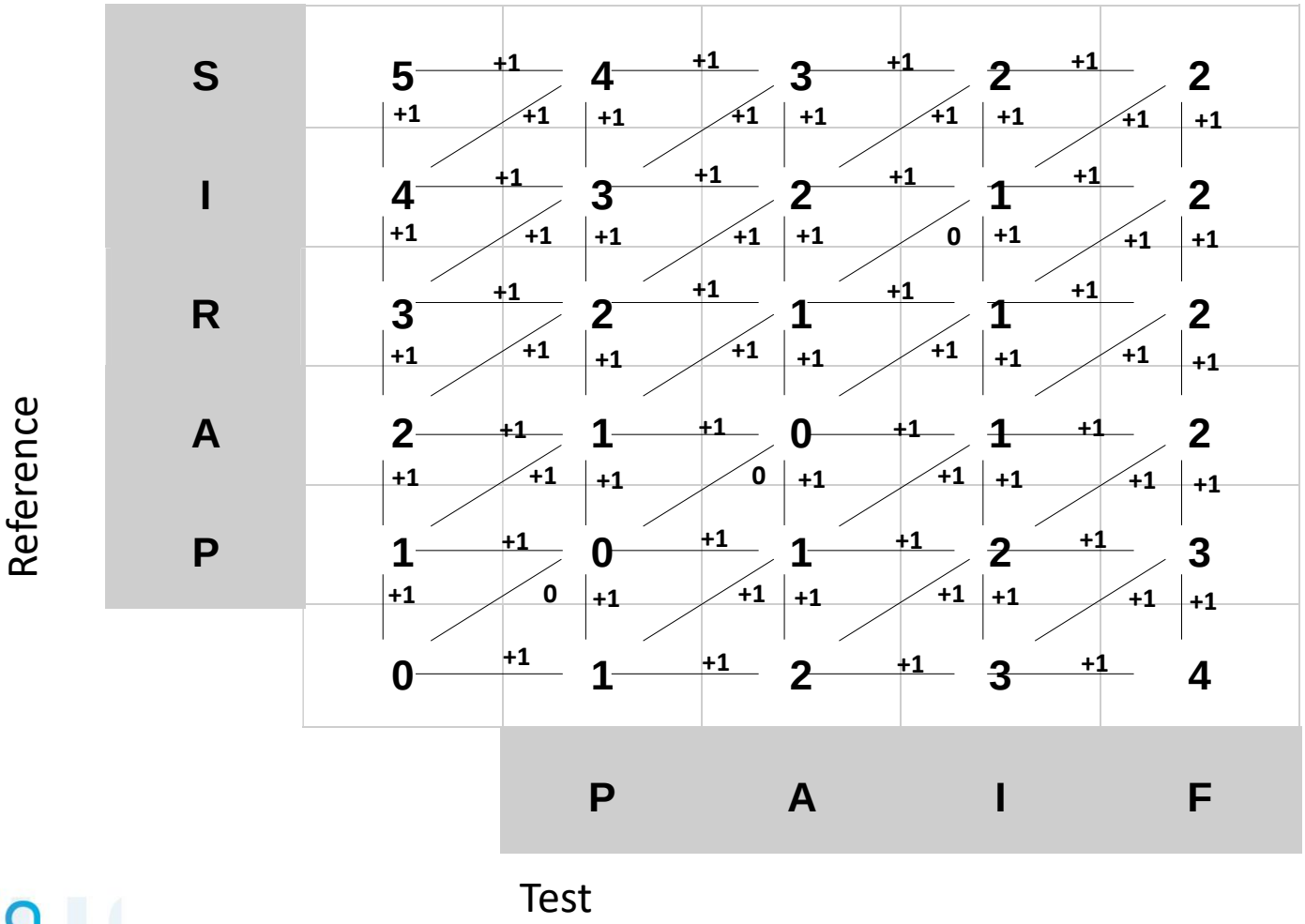
Dynamic programming example (graph)



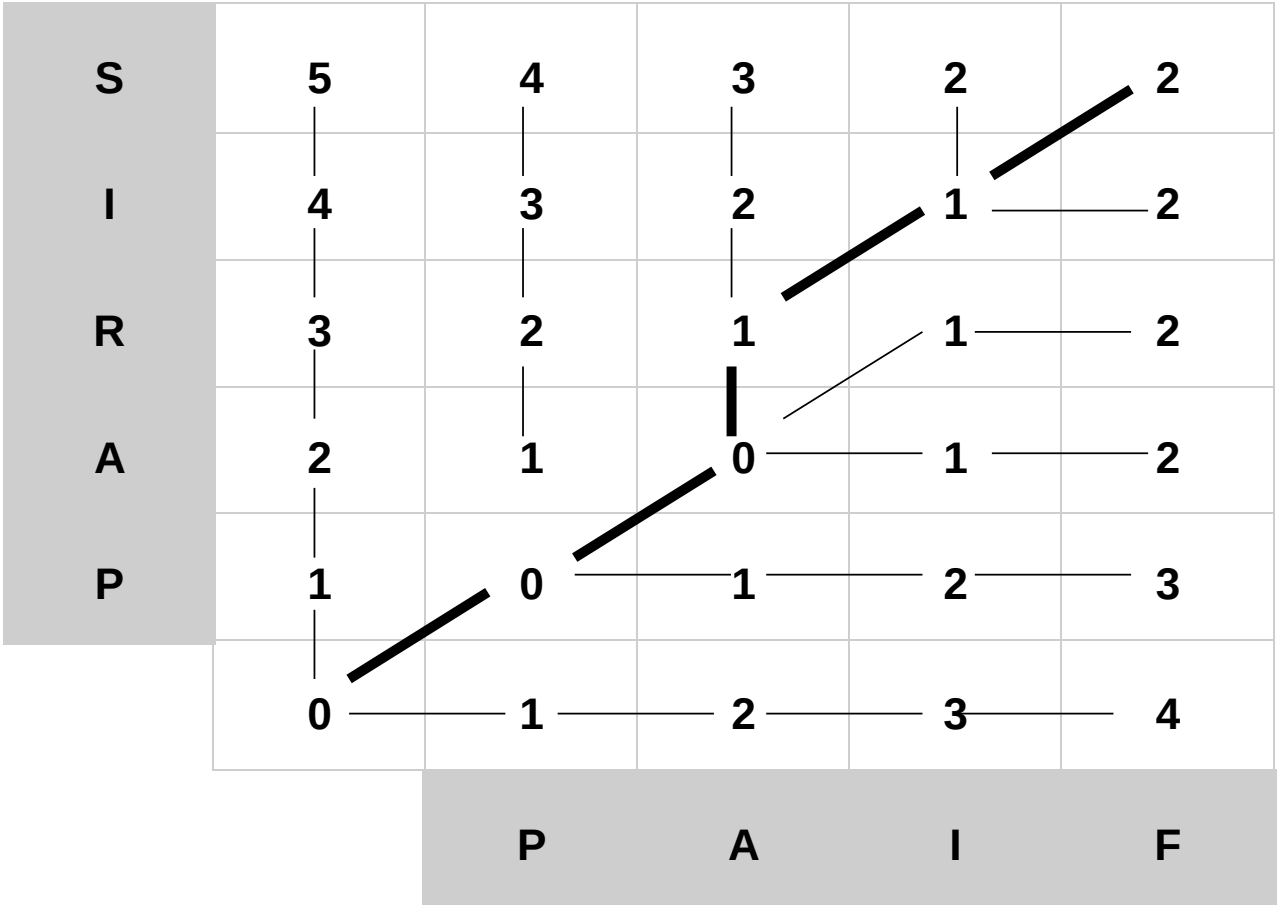
Dynamic programming example (costs)



Dynamic programming example (costs)



Dynamic programming example (alignment)



➔ (P ⇔ P) (A ⇔ A) (R deleted) (I ⇔ I) (S ⇔ F)

Terminology

- Dynamic programming

  General case (finding the best path in a graph)

- DTW : Dynamic Time Warping

  Used for aligning two acoustic forms (dealing with time modifications)

- Viterbi algorithm

  Associated to a probabilistic framework (for example hidden Markov models), finds the most likely path

- Baum-Welch algorithm

  Associated to a probabilistic framework (for example hidden Markov models), used for training model parameters (sum probabilities on all paths)