Artificial Intelligence
EDAF70

Lecture 11.2: Natural Language for Communication

Pierre Nugues

Lund University
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

February 27, 2019

Syntax

Grammar is the focus of natural language processing in the textbook (Russell and Norvig 2010, Chapter 23).
Two main (modern) traditions: constituent grammars (Chomsky, main advocate) and dependency grammars (Tesnière).
Constituent grammars are still dominant for English, although declining.
But they do not work well for Swedish, as well as many other languages.
Dependency grammars are more or less universal

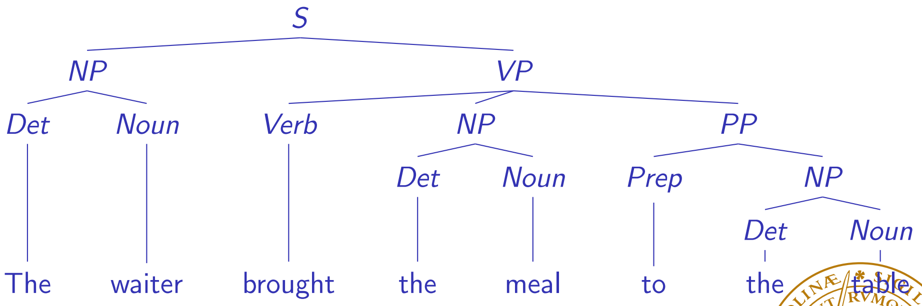
Constituents

The waiter brought the meal
The waiter brought the meal to the table
The waiter brought the meal of the day

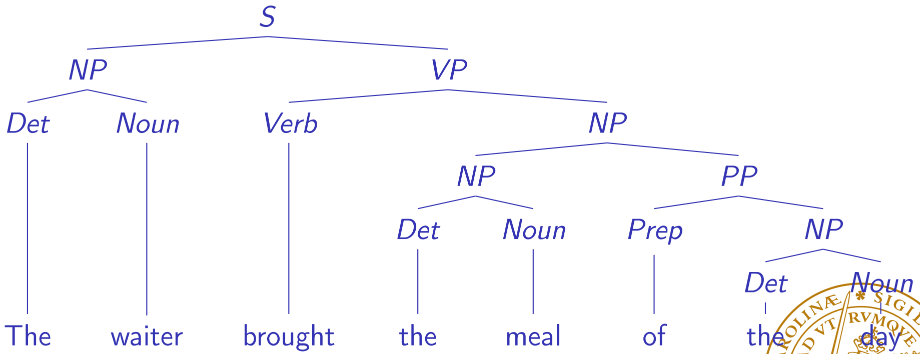
The waiter *brought* *the meal* *to* *the table*

The waiter *brought* *the meal* *of* *the day*

Syntactic Trees



Syntactic Trees



Lexicon (DCG)

```
noun --> [stench] ; [breeze] ; [glitter] ; [nothing] ;
  [wumpus] ; [pit] ; [pits] ; [gold] ; [east].
verb --> [is] ; [see] ; [smell] ; [shoot] ; [feel] ;
  [stinks] ; [go] ; [grab] ; [carry] ; [kill] ; [turn].
adjective --> [right] ; [left] ; [east] ; [south] ; [dead] ;
  [back] ; [smelly].
adverb --> [here] ; [there] ; [nearby] ; [ahead] ; [right] ;
  [left] ; [east] ; [south] ; [back].
pronoun --> [me] ; [you] ; ['I'] ; [it]; [she]; [he].
pnoun --> ['John'] ; ['Mary'] ; ['Boston'] ; ['UCB'] ;
  ['PAJC'].
article --> [the] ; [a] ; [an].
preposition --> [to] ; [in] ; [on] ; [near].
conjunction --> [and] ; [or] ; [but].
digit --> [0] ; [1] ; [2] ; [3] ; [4] ; [5] ; [6] ; [7]
```

Grammar Rules (DCG)

```
s --> np, vp. % I + feel a breeze
s --> s, conjunction, s.
np --> pronoun. %I
np --> pnoun.
np --> noun. %pits
np --> article, noun. %the + wumpus
np --> digit, digit. %3 4
np --> np, pp. %the wumpus + to the east
np --> np, rel_clause. %the wumpus + that is smelly
vp --> verb. %stinks
vp --> vp, np. %feel + a breeze
vp --> vp, adjective. %is + smelly
vp --> vp, pp. %turn + to the east
vp --> vp, adverb. %go + ahead
pp --> preposition, np. %to + the east
```

Parsing and Generation

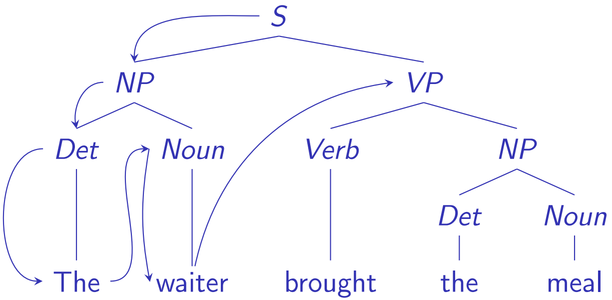
Parsing tells if a sentence is correct according to the grammar

```
?-s([the, wumpus, is, dead], []).
yes.
?- s([the, wumpus, that, stinks, is, in, 2, 2], []).
yes.
```

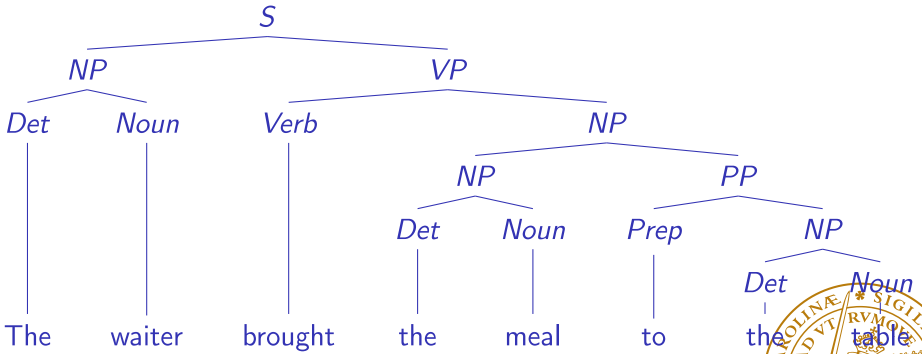
The parser can generate all the solutions

```
?- s(L, []).
L = [me, is] ;
L = [me, see] ;
L = [me, smell] ;
L = [me, shoot] ;
L = [me, feel] ;
L = [me, stinks] ;
```

The Prolog Search



Ambiguity


S
NP
VP
Det
Noun
Verb
NP
NP
PP
Det
Noun
Prep
NP
Det
Noun
The
waiter
brought
the
meal
to
the
table

Left-Recursive Rules

```
np --> np, pp.
```

The sentence:

The wumpus in the pit is dead

traps the parser in an infinite recursion.
We can use auxiliary symbols to remove left recursion:

```
npx --> det, noun.
np --> npx.
np --> npx, pp.
```

Variables

Overgeneration:

```
?- s(X, []).
X = [me, is] ;
X = [me, see] ;
X = [me, smell] ;
```

Solution: Add variables to differentiate between subject and object pronouns.

```
s --> np(s), vp.
np(Case) --> pronoun(Case).
pronoun(s) --> [you] ; ['I'] ; [it]; [she]; [he].
pronoun(o) --> [me] ; [you] ; [it].
```

Probabilistic Context-Free Grammars

$$P(T,S) = \prod_{rule(i)\,\text{producing}\,T} P(rule(i)).$$

where

$$P(lhs \rightarrow rhs_i | lhs) = \frac{Count(lhs \rightarrow rhs_i)}{\sum_j Count(lhs \rightarrow rhs_j)}.$$

An Example of PCFG

Rules			*P*	Rules			*P*
s	-->	np vp	0.8	det	-->	the	1.0
s	-->	vp	0.2	noun	-->	waiter	0.4
np	-->	det noun	0.3	noun	-->	meal	0.3
np	-->	det adj noun	0.2	noun	-->	day	0.3
np	-->	pronoun	0.3	verb	-->	bring	0.4
np	-->	np pp	0.2	verb	-->	slept	0.2
vp	-->	v np	0.6	verb	-->	brought	0.4
vp	-->	v np pp	0.1	pronoun	-->	he	1.0
vp	-->	v pp	0.2	prep	-->	of	0.6
vp	-->	v	0.1	prep	-->	to	0.4
pp	-->	prep np	1.0	adj	-->	big	1.0

Parse Trees of *Bring the meal of the day*

Parse trees

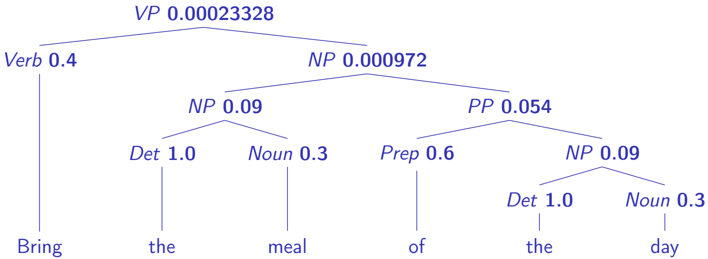
```
T1:  vp(verb(bring),
        np(np(det(the), noun(meal)),
           pp(prep(of), np(det(the), noun(day)))))
T2:  vp(verb(bring),
        np(np(det(the), noun(meal))),
        pp(prep(of), np(det(the), noun(day))))
```

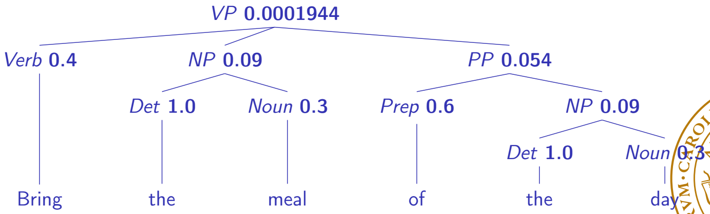
Computing the Probabilities

$$\begin{aligned}
&P(T_1, \text{Bring the meal of the day}) = \\
&P(vp \to v, np) \times P(v \to Bring) \times P(np \to np, pp) \times \\
&P(np \to det, noun) \times P(det \to the) \times P(noun \to meal) \times \\
&P(pp \to prep, np) \times P(prep \to of) \times P(np \to det, noun) \times \\
&P(det \to the) \times P(noun \to day) = \\
&0.6 \times 0.4 \times 0.2 \times 0.3 \times 1.0 \times 0.3 \times 1.0 \times 0.6 \times 0.3 \times 1.0 \times 0.3 = 0.00023328,
\end{aligned}$$

$$\begin{aligned}
&P(T_2, \text{Bring the meal of the day}) = \\
&P(vp \to v, np, pp) \times P(v \to Bring) \times P(np \to det, noun) \times \\
&P(det \to the) \times P(noun \to meal) \times P(pp \to prep, np) \times P(prep \to of) \times \\
&P(np \to det, noun) \times P(det \to the) \times P(noun \to day) = \\
&0.1 \times 0.4 \times 0.3 \times 1.0 \times 0.3 \times 1.0 \times 0.6 \times 0.3 \times 1.0 \times 0.3 = 0.0001944.
\end{aligned}$$

Computing the Probabilities





Semantic Parsing

Converts sentences to first-order logic or predicate-argument structures

Example:

Mr. Schmidt called Bill to

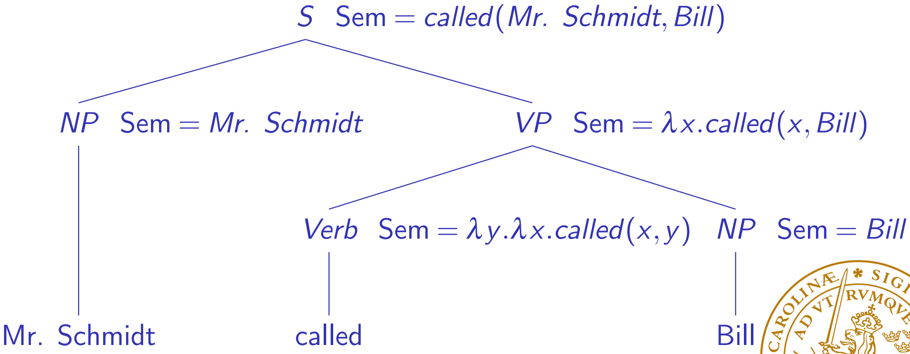
```
called('Mr. Schmidt', 'Bill').
```

Assumption: We can compose sentence fragments (phrases) into logical forms while parsing

This corresponds to the compositionality principle

Semantic Composition

Semantic composition can be viewed as a parse tree annotation



Getting the Semantic Structure

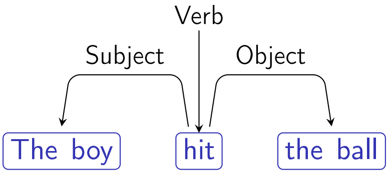
Bill rushed `rushed('Bill').`

The verb *rushed* is represented as a lambda expression: $\lambda x.rushed(x)$
Beta reduction: $\lambda x.rushed(x)(Bill) = rushed(Bill)$
Lambda expressions are represented in Prolog as `X^rushed(X).`

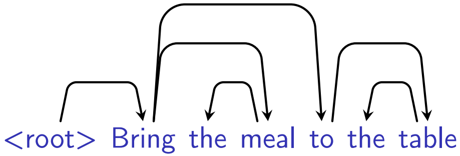
The patron ordered a meal	`ordered(patron, meal)`
ordered a meal	`X^ordered(X, meal)`
ordered	`Y^X^ordered(X, Y)`

The Current Approach: Dependencies

A graph of dependencies and functions:



Conventions: Each word has a head and the main word is linked to an artificial root:



Parsing Dependencies

Generate all the pairs:

Which sentence root?

Bring
the
meal
to
the
table

Which head?
meal

Algorithms: Extensions to shift-reduce or graph optimization trained on annotated corpora.
Corpora: `https://universaldependencies.org/`

Machine Translation

Natural language processing was born with machine translation

Massive advance when the US government decided to fund large-scale translation programs to have a quick access to documents written in Russian

IBM teams pioneered statistical models for machine translation in the early 1990s

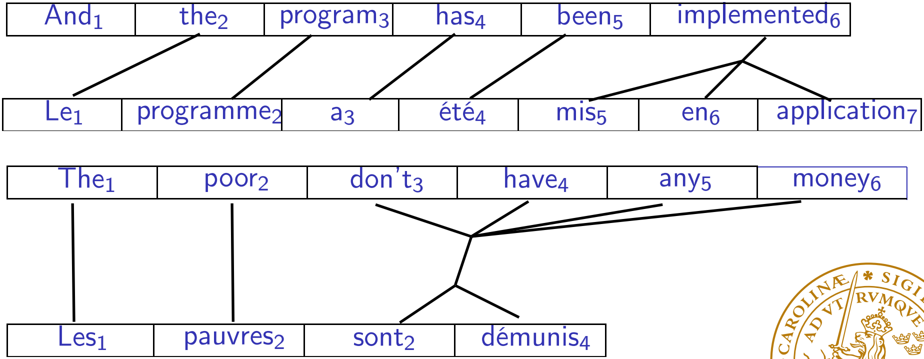
Their work that used the English (e) and French (f) parallel versions of the Canadian Hansards is still the standard reference in the field.

Parallel Corpora (Swiss Federal Law)

German	French	Italian
Art. 35 Milchtransport	**Art. 35 Transport du lait**	**Art. 35 Trasporto del latte**
1 Die Milch ist schonend und hygienisch in den Verarbeitungsbetrieb zu transportieren. Das Transportfahrzeug ist stets sauber zu halten. Zusammen mit der Milch dürfen keine Tiere und milchfremde Gegenstände transportiert werden, welche die Qualität der Milch beeinträchtigen können.	1 Le lait doit être transporté jusqu'à l'entreprise de transformation avec ménagement et conformément aux normes d'hygiène. Le véhicule de transport doit être toujours propre. Il ne doit transporter avec le lait aucun animal ou objet susceptible d'en altérer la qualité.	1 Il latte va trasportato verso l'azienda di trasformazione in modo accurato e igienico. Il veicolo adibito al trasporto va mantenuto pulito. Con il latte non possono essere trasportati animali e oggetti estranei, che potrebbero pregiudicarne la qualità.

Alignment (Brown et al. 1993)

Canadian Hansard



Machine Translation Algorithms

A statistical model:

$$P(f, d|e) = \prod_i P(f_i|e_i)P(d_i),$$

where d measures the distortion, how much reassembling is needed from English to French.

Distortion has the form of a right-to-left or left-to-right shift.

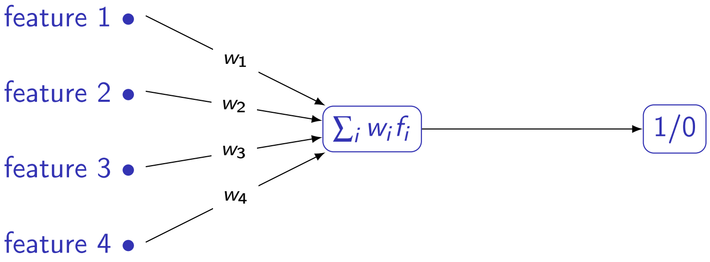
Steps to build a machine translation system:

- Build parallel corpora
- Segment and align sentences
- Align phrases
- Extract distortions
- Improve estimates

Recently, recurrent neural network architectures improved considerably the performance of machine translation.

Neural Networks: Representation

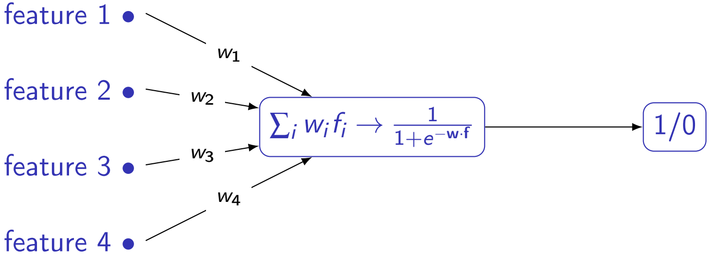
Another representation of the perceptron:



The base network: An input layer and an output layer

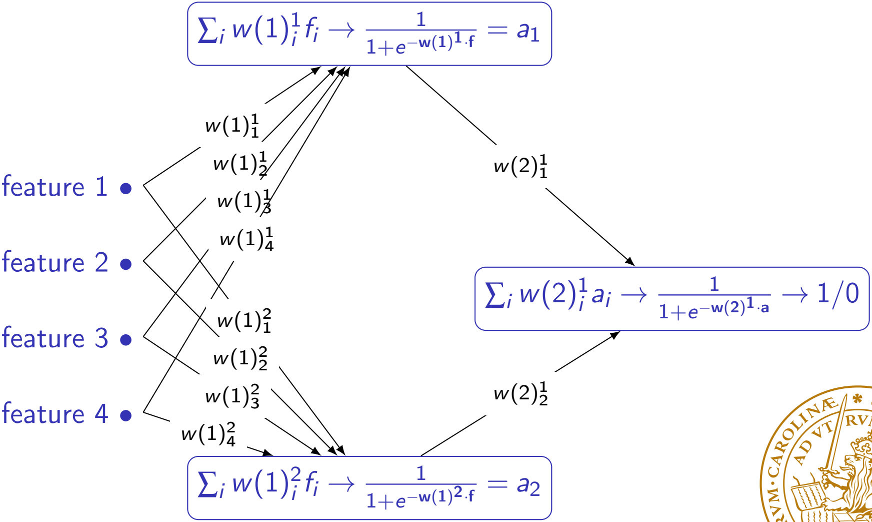
Neural Networks: Activation Function

And logistic regression:



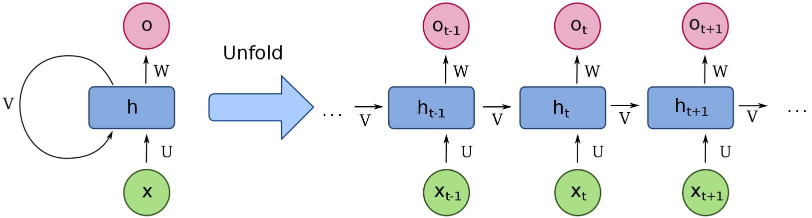
The logistic function is the activation function of the node

Neural Networks: Hidden Layers



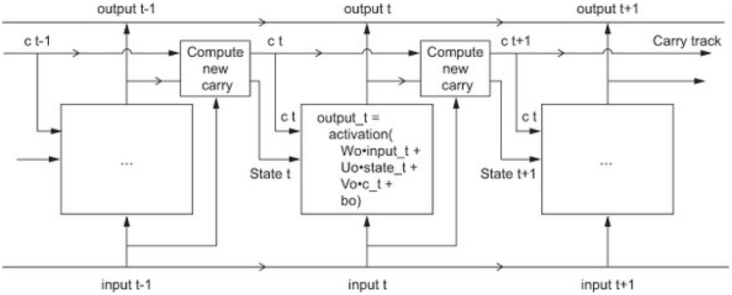
Demonstration: http://playground.tensorflow.org/

Recurrent Neural Networks



Source: Wikipedia

Long Term Short Term Memory (LSTM) Networks



Source: François Chollet, Deep Learning for Python, Manning, 2018, page 204, an excellent book

See also:

- https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning.html

Speech Recognition

Conditions to take into account:

- Number of speakers
- Fluency of speech.
- Size of vocabulary
- Syntax
- Environment

Structure of Speech Recognition

Words:

$$W = w_1, w_2, ..., w_n.$$

Acoustic symbols:

$$A = a_1, a_2, ..., a_m,$$

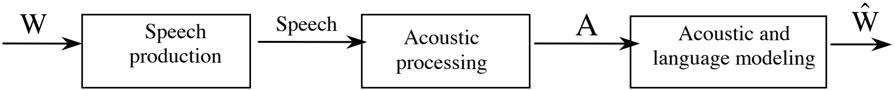
$$\hat{W} = \arg\max_W P(W|A).$$

Using Bayes' formula,

$$P(W|A) = \frac{P(A|W)P(W)}{P(A)}.$$

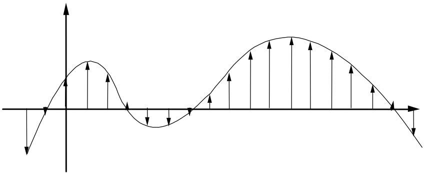
Two-Step Recognition

$$\hat{W} = \arg\max_{W} P(A|W)P(W).$$

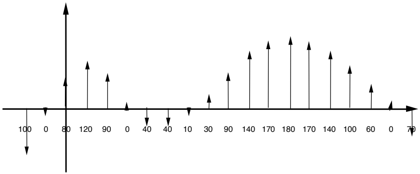


Signals

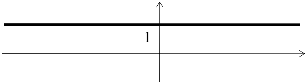
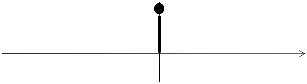
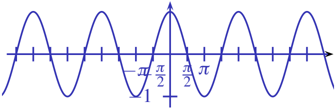
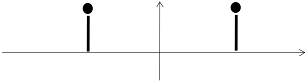
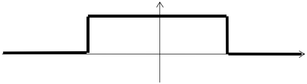
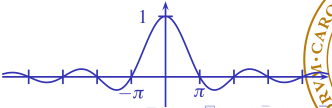
Sampling



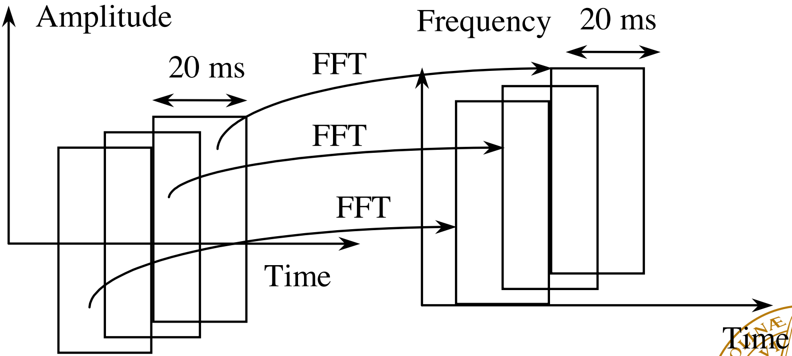
Digitization



Fourier Transforms

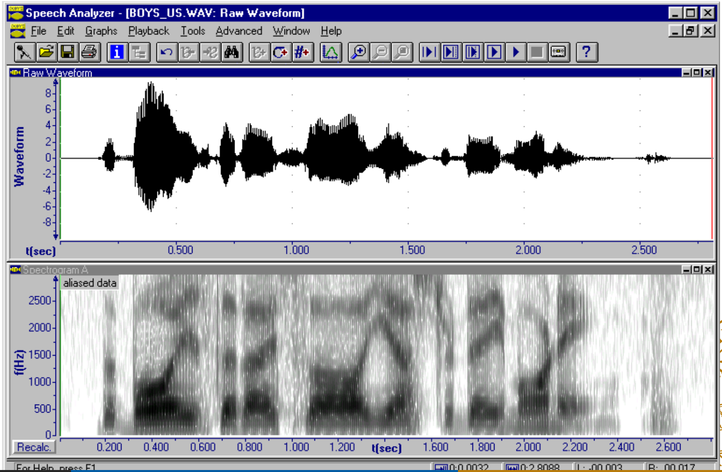
Time domain	**Frequency domain (Fourier Transforms)**
Unit constant function: $f(x) = 1$	Delta function, perfect impulse at 0: $\delta(x)$
	
Cosine: $\cos(2\pi\omega x)$	Shifted deltas: $\frac{\delta(x+\omega)+\delta(x-\omega)}{2}$
	
Square pulse : $w_a(x) = \begin{cases} 1 & -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 0 & elsewhere \end{cases}$	$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$
	

Speech Spectrograms



Speech Signals

The boys I saw yesterday morning



Speech Parameters

Recognition devices derive a set of acoustic parameters from speech frames.
Parameters should be related to “natural” features of speech: voiced or unvoiced segments.
A simple parameter giving a rough estimate of it: the energy: the darker the frame, the higher the energy.

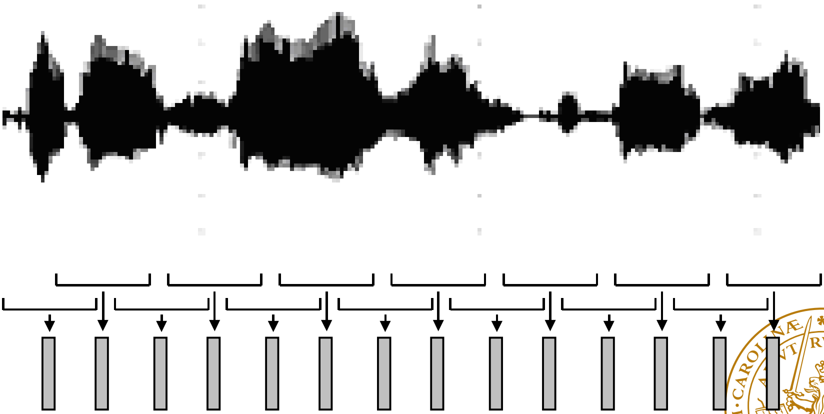
$$E(F_k) = \sum_{n=m}^{m+N-1} s^2(n).$$

Linear prediction coefficients:

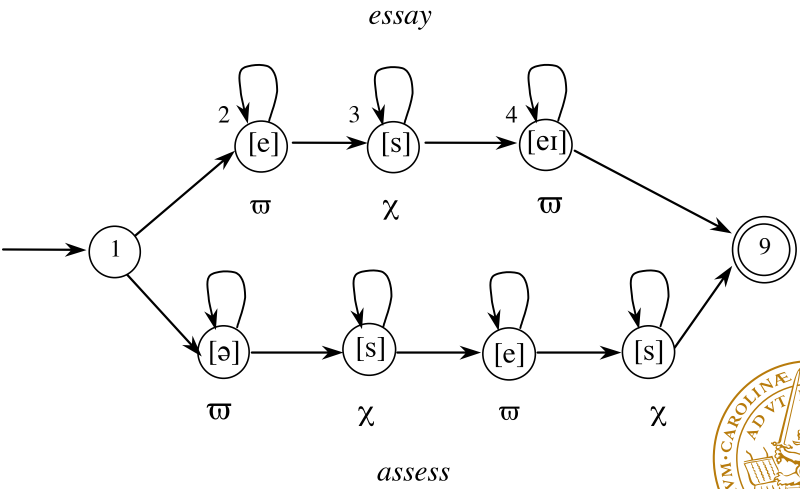
$$\hat{s}(n) = a(1)s(n-1) + a(2)s(n-2) + a(3)s(n-3) + ... + a(m)s(n-m).$$

Extraction of Speech Parameters

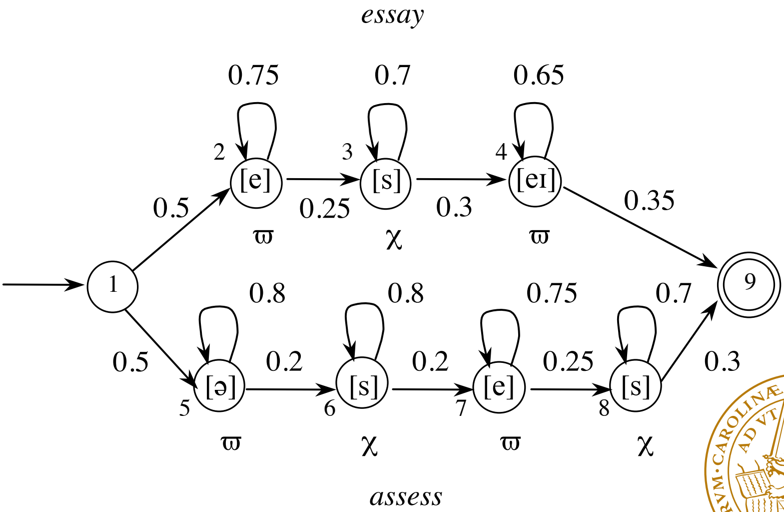
Features are extracted every 10 ms over a 20 s frame



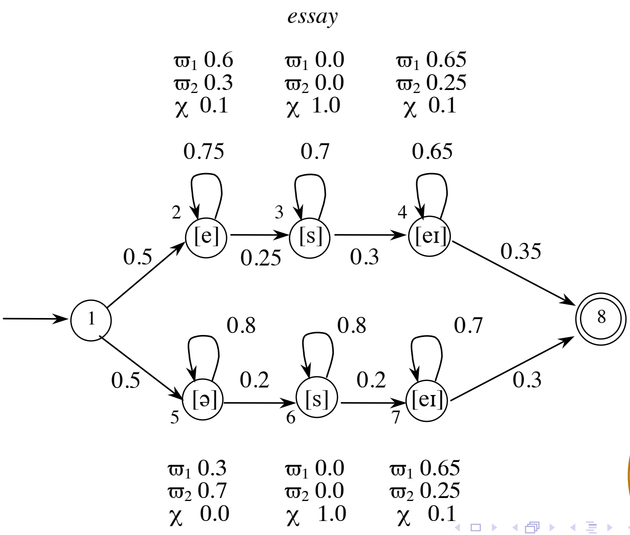
Automata



Markov Chains



Hidden-Markov Models



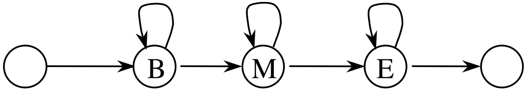
Solving Problems with Hidden-Markov Models

Given a hidden-Markov model, the main problems to solve are to:

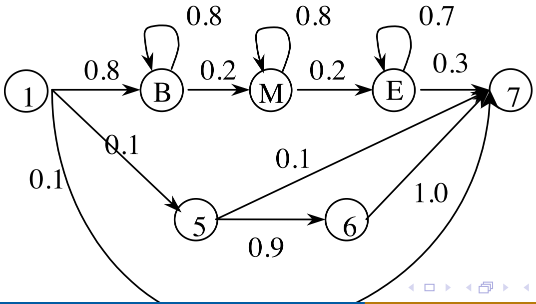
- Estimate the probability of an observed sequence. It corresponds to the sum of all the paths producing the observation. It is solved using the forward algorithm.
- Determine the most likely path of an observed sequence. It is a decoding problem. It is solved using the Viterbi algorithm.
- Determine (learn) the parameters given a set of observations. It is used to build models to recognize speech. It is solved using the forward-backward algorithm.

HMM and Phones

Modeling phones:
Simple model



A more complex model due to Lee



Neural Networks for Speech Recognition(I)

From 2015-2016, neural network architectures started to overtake HMM.
Most current systems use variants of recurrent neural networks.
A historical model from Waibel et al., Phoneme recognition using time-delay neural networks, 1989.

- Three phonemes B, D, and G
- An input vector consists of 16 melscale coefficients from a Fourier transform of a speech window of 10 ms: Energy at certain frequencies
- The context is modeled as a sequence of three such input vectors.
- Two hidden layers

Neural Networks for Speech Recognition(II)

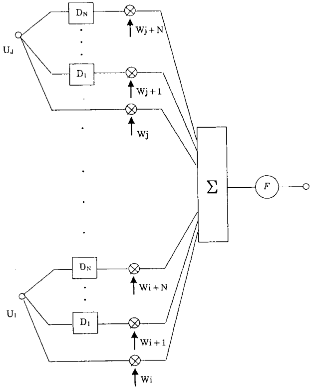


Fig. 1. A Time-Delay Neural Network (TDNN) unit.

From Waibel et al., Phoneme recognition using time-delay neural networks, IEEE Transactions of Acoustics, Speech, and Signal Processing, 37(3) 1989.

Neural Networks for Speech Recognition (III)

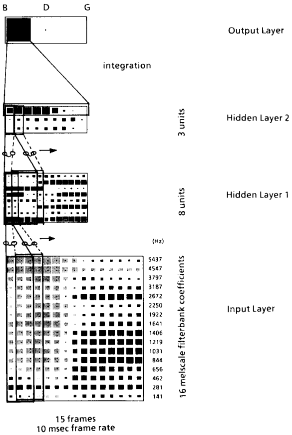


Fig. 2. The architecture of the TDNN.

From Waibel et al., Phoneme recognition using time-delay neural networks, IEEE Transactions of Acoustics, Speech, and Signal Processing, 37(3) 1989.