Automatic speech recognition

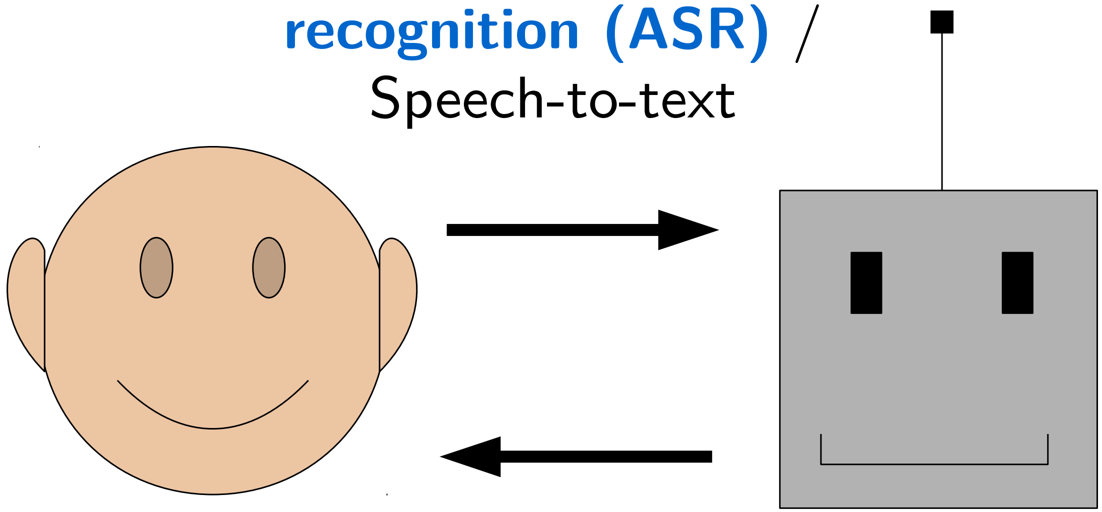
Gwénolé Lecorvé – gwenole.lecorve@irisa.fr

Research in Computer Science (SIF) master

Institut de Recherche en Informatique et Systèmes Aléatoires

Spoken interaction

Automatic speech recognition (ASR) / Speech-to-text

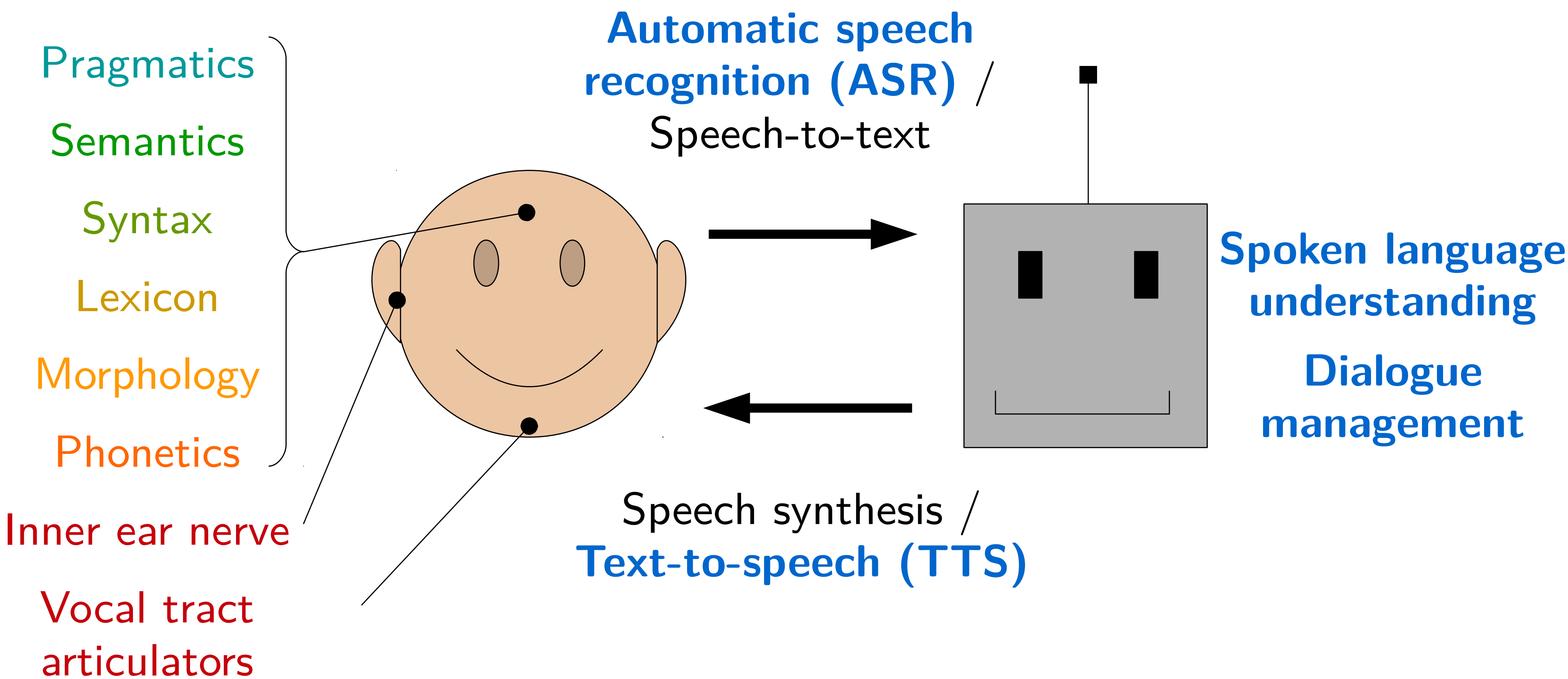


Spoken language understanding

Dialogue management

Speech synthesis / **Text-to-speech (TTS)**

Spoken interaction



Spoken interaction

G. Lecorvé

Automatic speech recognition (ASR) / Speech-to-text

Pragmatics

Semantics

Syntax

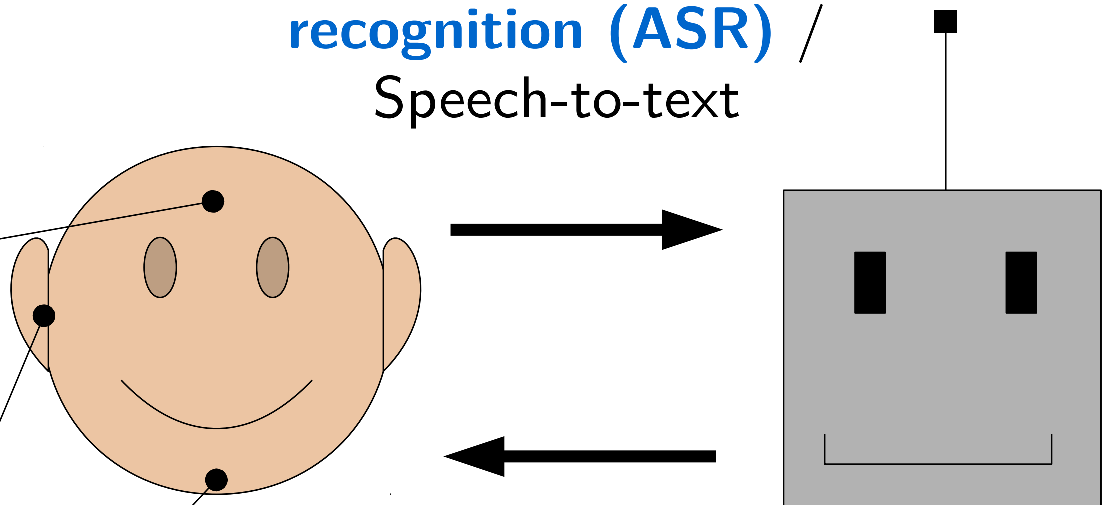
Lexicon

Morphology

Phonetics

Inner ear nerve

Vocal tract articulators



C. Raymond

Spoken language understanding

Dialogue management

Speech synthesis / **Text-to-speech (TTS)**

G. Lecorvé

Outline

Introduction and definitions

Reading:

- Jurafsky and Martin (2008). Speech and Language Processing (2nd ed.)

What is speech recognition?

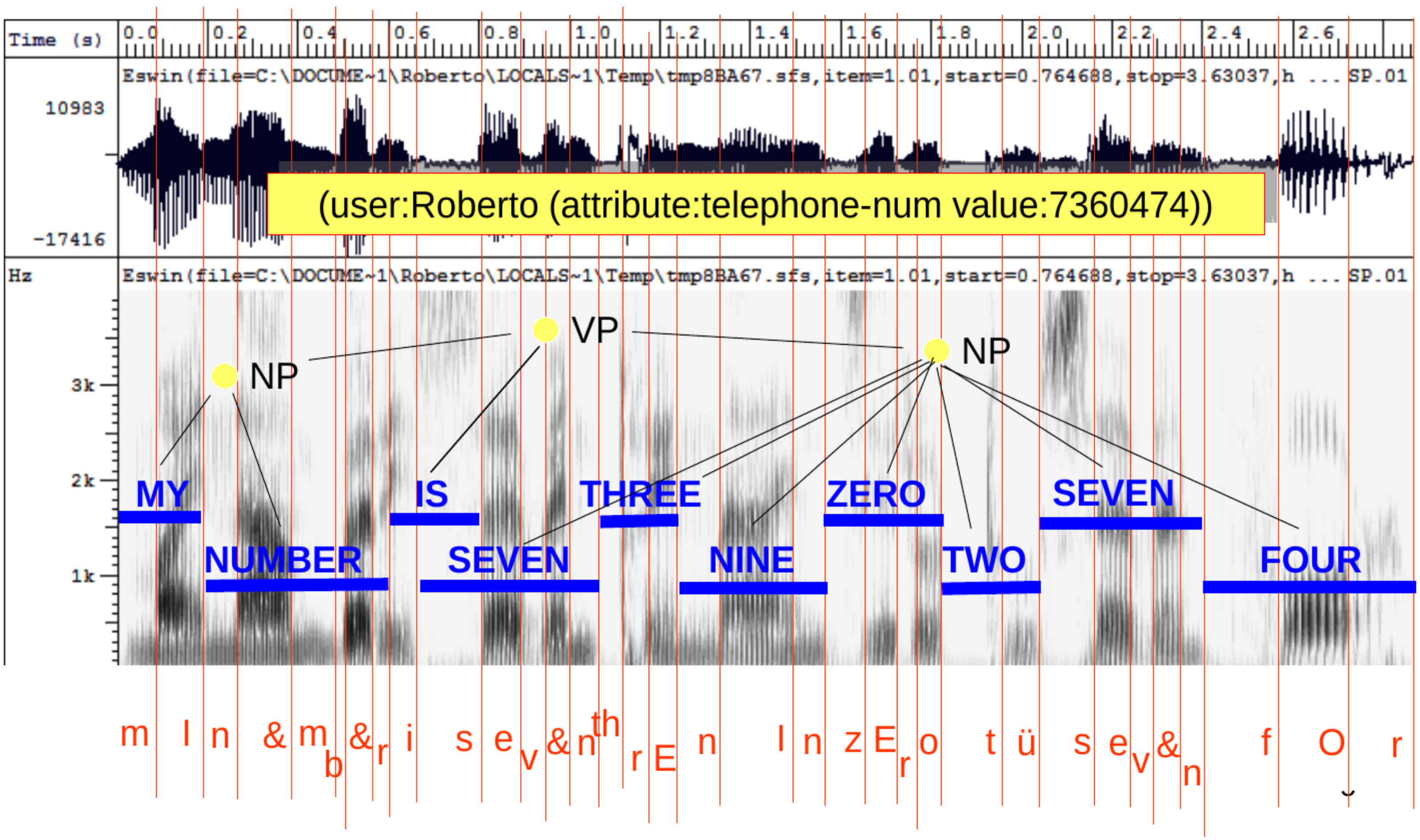
- Transform raw audio into a sequence of words
  - No meaning

    $\rightarrow$ "Recognize speech" $\sim$ "Wreck a nice beach"

    $\rightarrow$ "Barack Obama" $\sim$ "Barraque aux Bahamas"
- Related tasks
  - Speaker diarization/recognition: Who spoke when?
  - Spoken langage understanding: What's the meaning?
  - Sentiment analysis, opinion mining: How does the speaker feel/think?

Difficulties

- Hierarchical problem?

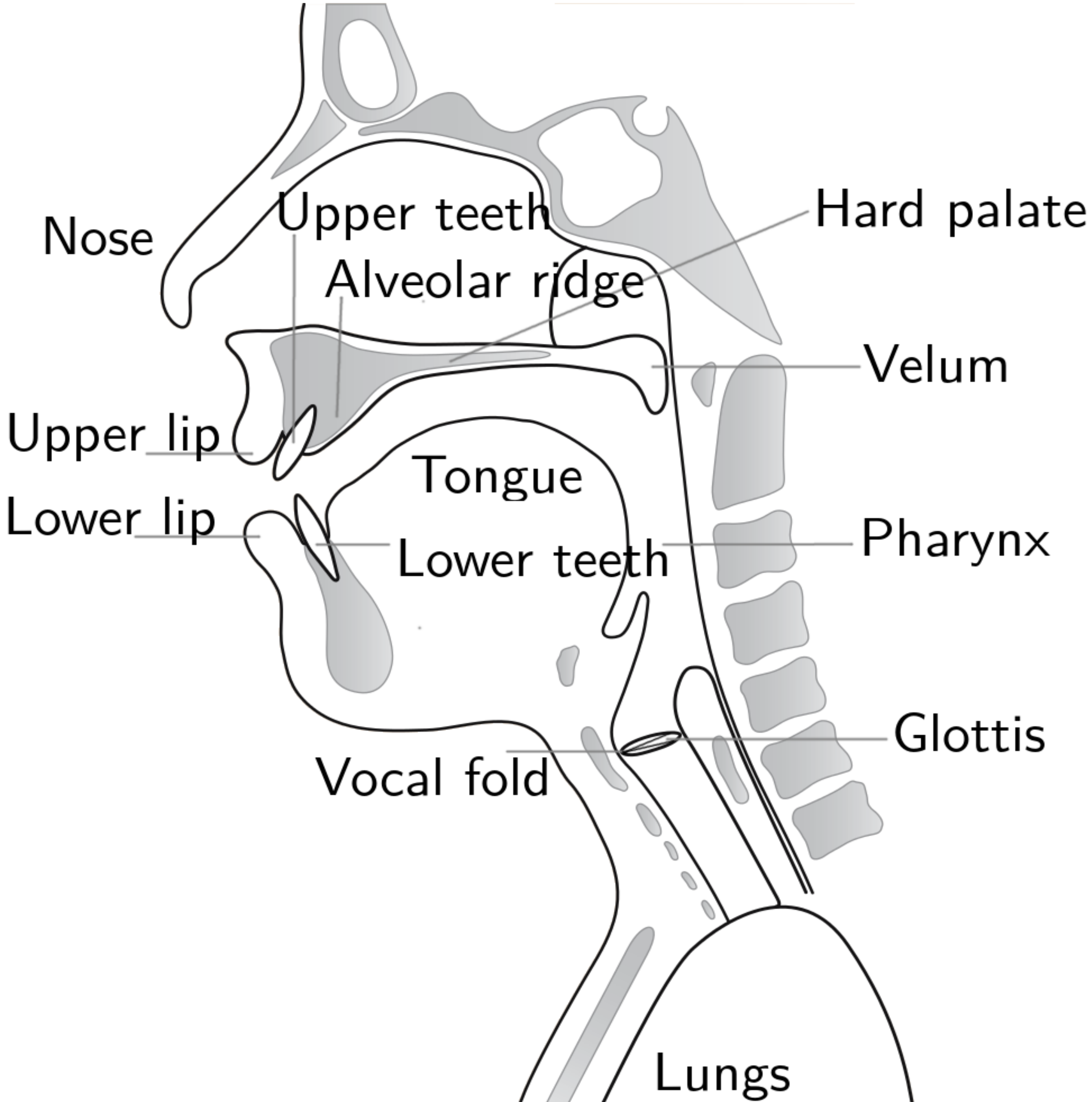


(Source: Julia Hirschberg)

Difficulties

- Not that simple because lots of variability
  - Acoustics
    - Intra-speaker variability, inter-speaker variability
    - Noise, reverberation, etc.
  - Phonetics
    - Co-articulation, elisions, etc.
    - Word confusability
  - Linguistics
    - Word variations
    - Vocabulary size
    - Polysemy
    - Elipses, anaphore, etc.

Speech production

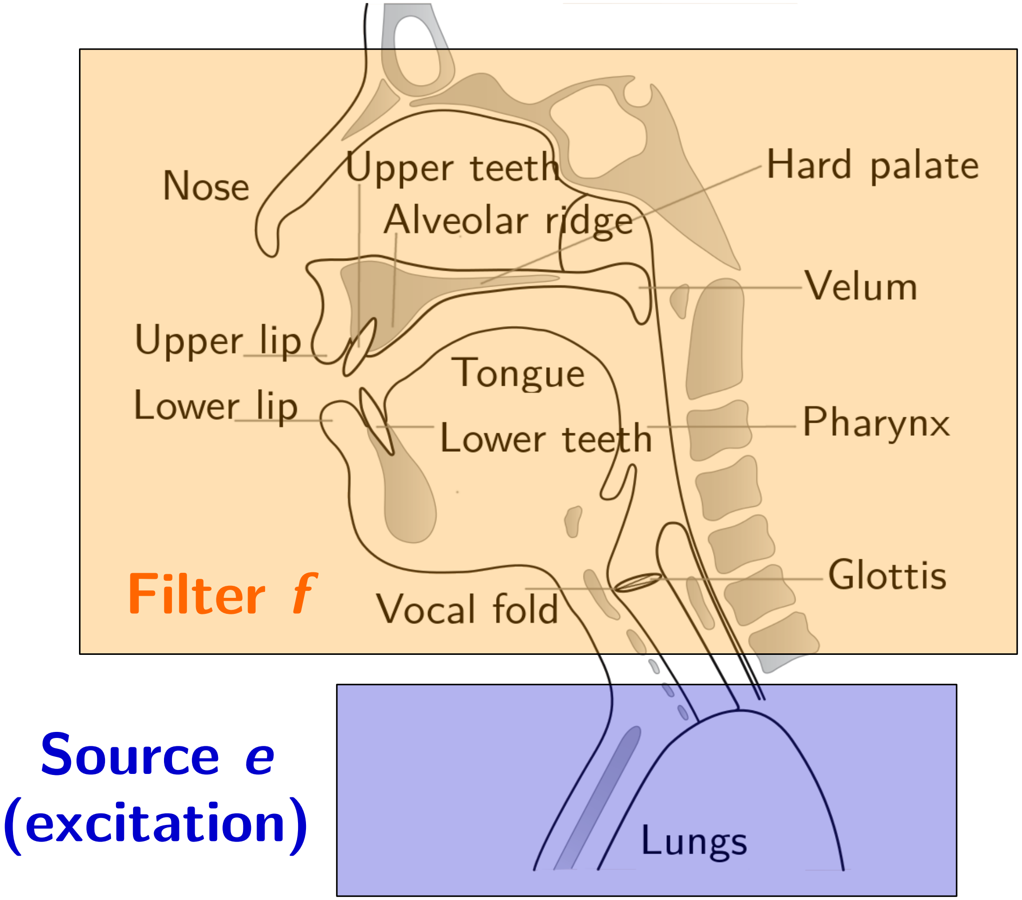


Speech production

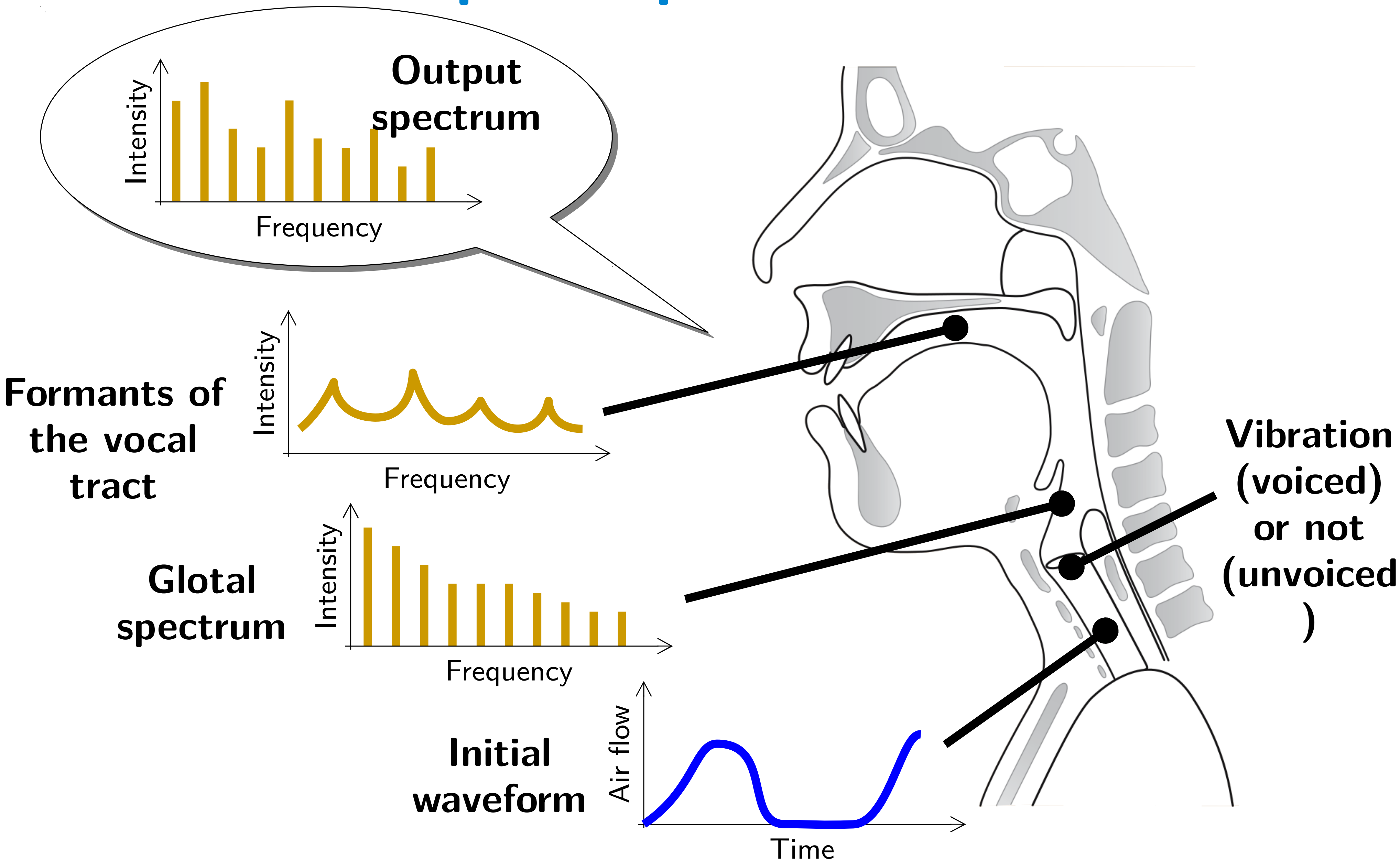
- Source filter model (Fant, 1960)
- Signal $s = f * e$

$$s(t) = \int_{-\infty}^{+\infty} e(t) f(t - \tau) d\tau$$

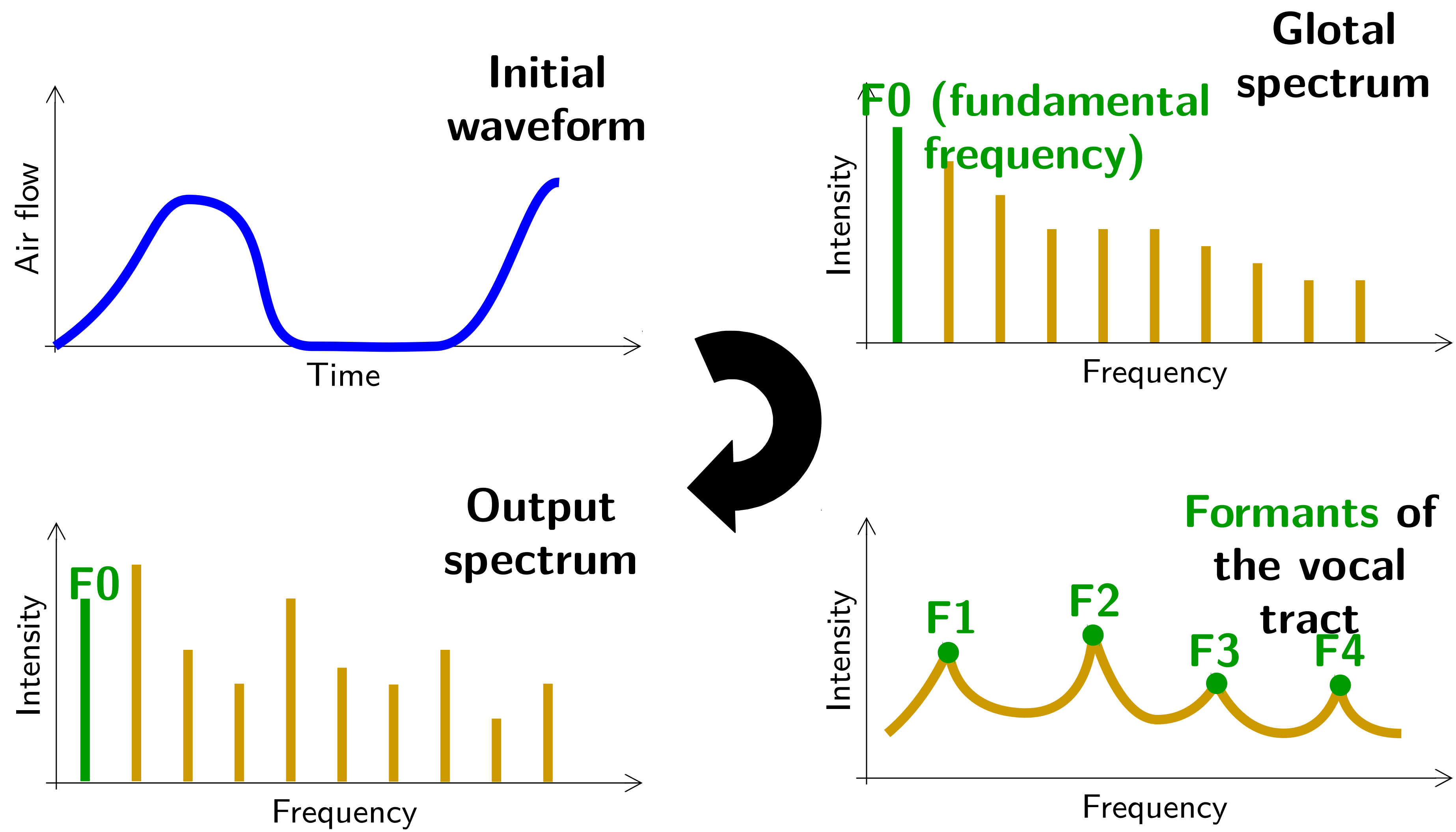
(assuming f is linear and time-independent)



Speech production



Speech production



Phonetics and phonology

- Spoken words are made of **phonemes/phones**
  - "French" → / f ɹ ɛ n t ʃ /
  - "français" → / f ʁ ɑ̃ s ɛ /
- Acoustic view
  - Phone
  - Realized
- Phonological view
  - Phoneme
  - Symbolic

International Phonetic Alphabet (IPA)

[f o ʊ̯ n]

/ f o ʊ n i m /

Phonetics and phonology

- 1 phoneme = voiceness/unvoiceness
  + position of articulators
- All phonemes = set of **elementary** sounds in a **langage**
  - Language-dependent (ɹ ≠ ʁ)
  - Elementary : principle of minimal pairs
    - "kill" versus "kiss"
    - "pat" versus "bat"
- Allophones = free variants of a phonemes
  - No minimal pair
  - "père" → [pɛr], [pɛʀ] or [pɛʁ]

Phonetics and phonology

Consonant phonemes of French

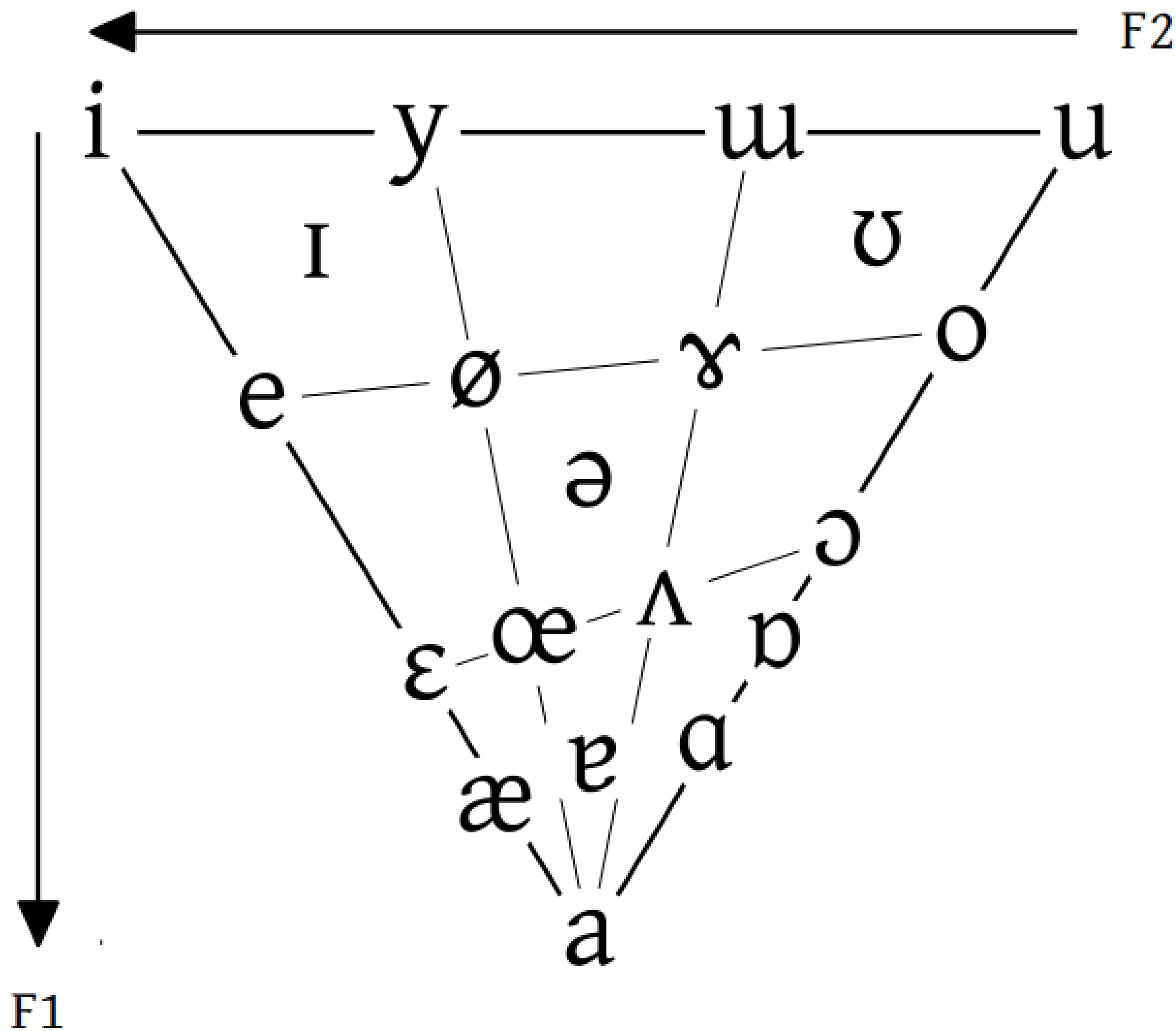
<table>
<tr><th colspan="2"></th><th>Labial</th><th>Dental/ Alveolar</th><th>Palatal</th><th>Velar</th><th>Uvular</th></tr>
<tr><th colspan="2">Nasal</th><td>m</td><td>n</td><td>ɲ</td><td>(ŋ)</td><td></td></tr>
<tr><th rowspan="2">Stop</th><th>voiceless</th><td>p</td><td>t</td><td></td><td>k</td><td></td></tr>
<tr><th>voiced</th><td>b</td><td>d</td><td></td><td>ɡ</td><td></td></tr>
<tr><th rowspan="2">Fricative</th><th>voiceless</th><td>f</td><td>s</td><td>ʃ</td><td>(x)</td><td></td></tr>
<tr><th>voiced</th><td>v</td><td>z</td><td>ʒ</td><td></td><td rowspan="2">ʁ</td></tr>
<tr><th rowspan="2">Approximant</th><th>plain</th><td></td><td>l</td><td>j</td><td></td></tr>
<tr><th>labial</th><td></td><td></td><td>ɥ</td><td>w</td><td></td></tr>
</table>

Phonetics and phonology

Vowel phonemes in Standard French

		Front		Central	Back
		unrounded	**rounded**		
Close	**oral**	i	y		u
Close-mid		e	ø	ə	o
Open-mid		ɛ (ɛː)	œ		ɔ
	nasal	ɛ̃	(œ̃)		ɔ̃
Open					ɑ̃
	oral	(a)		a	(ɑ)

Phonetics and phonology



Linguistics

- Word
  - Sequence of graphemes (symbolic view)
  - Morphemes: "recognition" = "re" + "cogni" + "tion"
  - Morpho-syntax: Part Of Speech (POS)
    - Grammatical class : Noun, verb, etc.
    - Flexional information : Singular/plural, gender, etc.
  - Syntax
    - Function : subject, object, etc.
    - Shallow, deep parsing (compound structures)
  - Meaning

  $\rightarrow$ Representation?

Linguistics

- Vocabulary = set of words in a
  - Task
  - Language
  - Several languages
- Syntax
  - None: isolated words
  - Grammar
  - Free

$\left.\begin{array}{l}\text{Grammar}\\ \text{Free}\end{array}\right\}$ **Continuous** speech recognition

$\left.\begin{array}{l}\text{Language}\\ \text{Several languages}\\ \text{Syntax}\\ \text{None: isolated words}\\ \text{Grammar}\\ \text{Free}\end{array}\right\}$ **Large vocabulary continuous** speech recognition (LVCSR)

Evaluation: Word Error Rate (WER)

- **Reference:** manual transcript `the lazy dog jumps`
- **Hypothesis:** ASR output `amazing dog jumps`
- Alignment

```
the     lazy dog jumps
*** amazing dog jumps
```

- Editings `Del     Sub`
- Score: $\mathrm{WER} = \dfrac{\mathsf{N}_{\mathrm{Ins}} + \mathsf{N}_{\mathrm{Del}} + \mathsf{N}_{\mathrm{Sub}}}{|\mathrm{Reference}|}$ Edit distance
  - Perfect = 0%
  - can be > 100% (many insertions)

`(0+1+1)/4 =` **50%**

Evaluation: Word Error Rate (WER)

- Word alignment: Wagner-Fischer algorithm (dynamic programming)
  - 3 costs: insertion, deletion, substitution

jumps	4	4	3	**2** (↙)
dog	3	3	**2** (↙)	3
lazy	2	**2** (↙)	2	3
the	**1** (↓)	1	2	3
<s>	**0**	1	2	3
	<s>	amazing	dog	jumps

→ All errors may not harm the same (w.r.t. task)

Statistical (historical) approach

Reading:

- Jurafsky and Martin (2008). Speech and Language Processing (2nd ed.)
- Jelinek (1998). Statistical Methods for Speech Recognition

Formalisation statistique

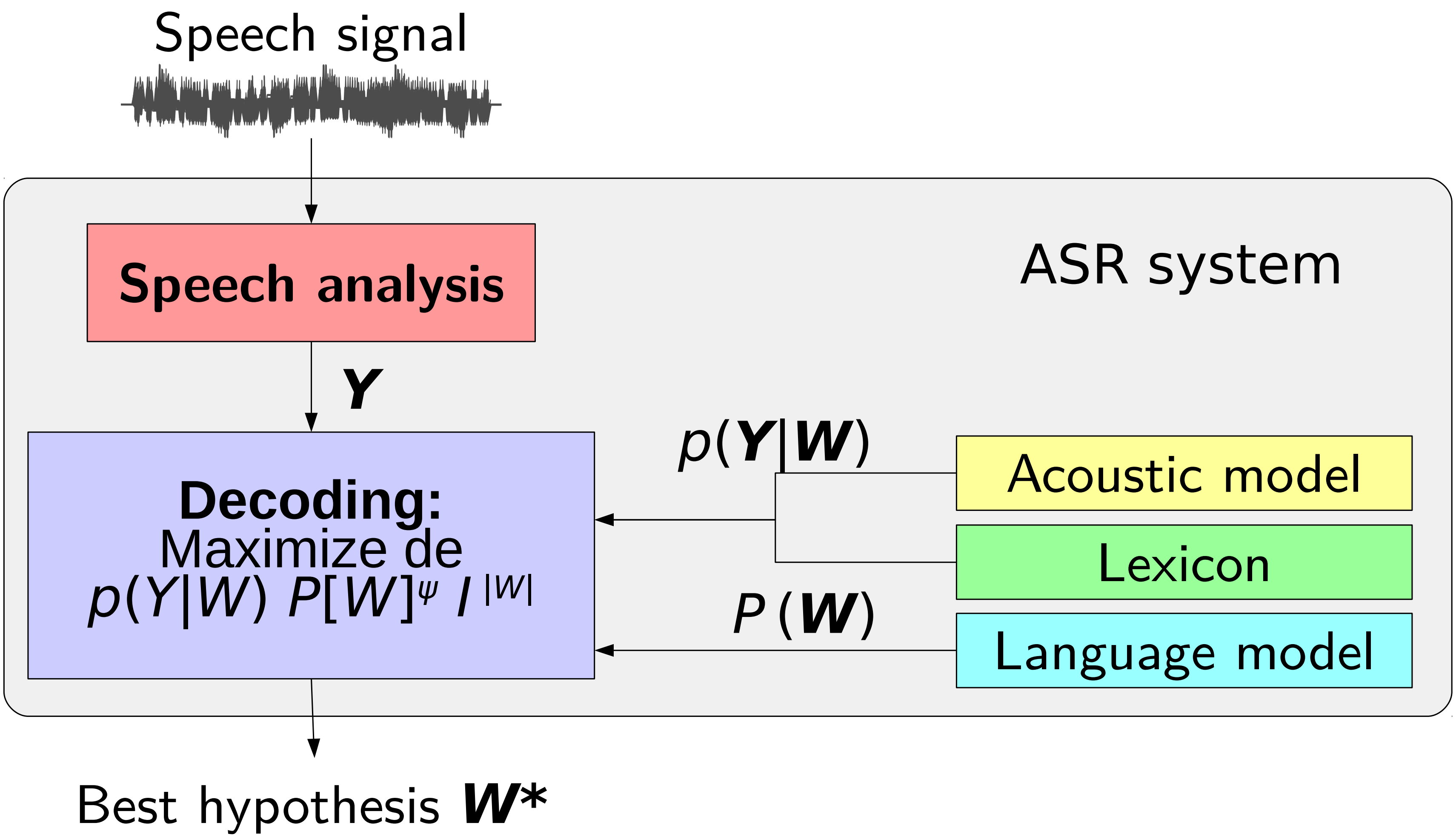
- **Y** : sequence of acoustif features
- **W** : sequence of words (of the voabulary)

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \mathrm{P}(\mathbf{W}|\mathbf{Y})$$

$$= \arg\max_{\mathbf{W}} \frac{\mathrm{p}(\mathbf{Y}|\mathbf{W})\ \mathrm{P}(\mathbf{W})}{\mathrm{p}(\mathbf{Y})}$$

$$= \underbrace{\boxed{\arg\max_{\mathbf{W}}}}_{\text{Search space } = \text{ f(Vocabulary)}} \underbrace{\boxed{\mathrm{p}(\mathbf{Y}|\mathbf{W})}}_{\text{Acoustic model}} \underbrace{\boxed{\mathrm{P}(\mathbf{W})}}_{\text{Language model}}$$

Steps and components



Generative view

Language model
$P(\boldsymbol{W})$

↓

Transcript hypothesis
$\boldsymbol{W} = w_1 \dots w_N$

↓

Pronunciation dictionary
$\boldsymbol{W} \rightarrow \boldsymbol{H}$

↓

Phoneme sequence
$\boldsymbol{H} = p_1 \dots p_L$

↓

Acoustic model
$p(\boldsymbol{Y}|\boldsymbol{H})$

↓

Acoustic features
$\boldsymbol{Y} = y_1 \dots y_T$

↓

Signal

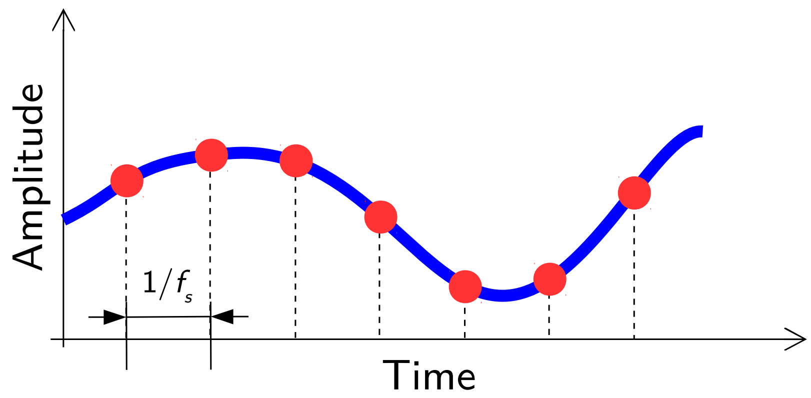
Speech analysis

Reading:

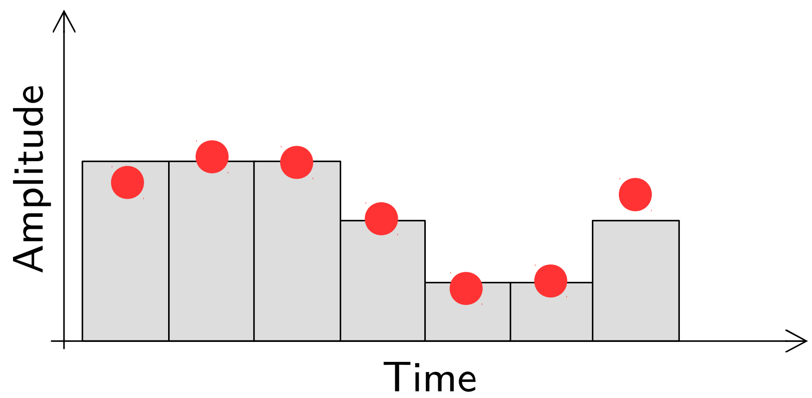
- Han et al. (2006). An efficient MFCC extraction method in speech recognition. IEEE International Symposium.
- Hermansky et al. (1992). RASTA-PLP speech analysis technique. In Proc. ICASSP.
- Hermansky et al. (2000). Tandem connectionist feature extraction for conventional HMM systems. In Proc. ICASSP

Sampling and quantization

- Sampling
  - Usual resolution $f_s = 8\text{kHz-}16\text{kHz}$

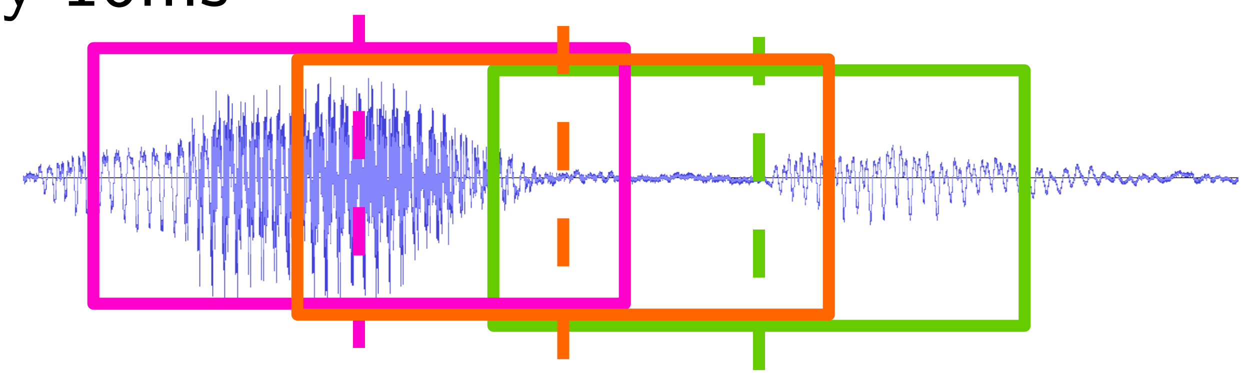


- Quantization
  - 8 bits / sample



Windowing, frames

- 1 frame = window of samples
- Overlap across frames
  - 32ms span (256 samples for $f_s = 8$kHz)
  - Every 10ms

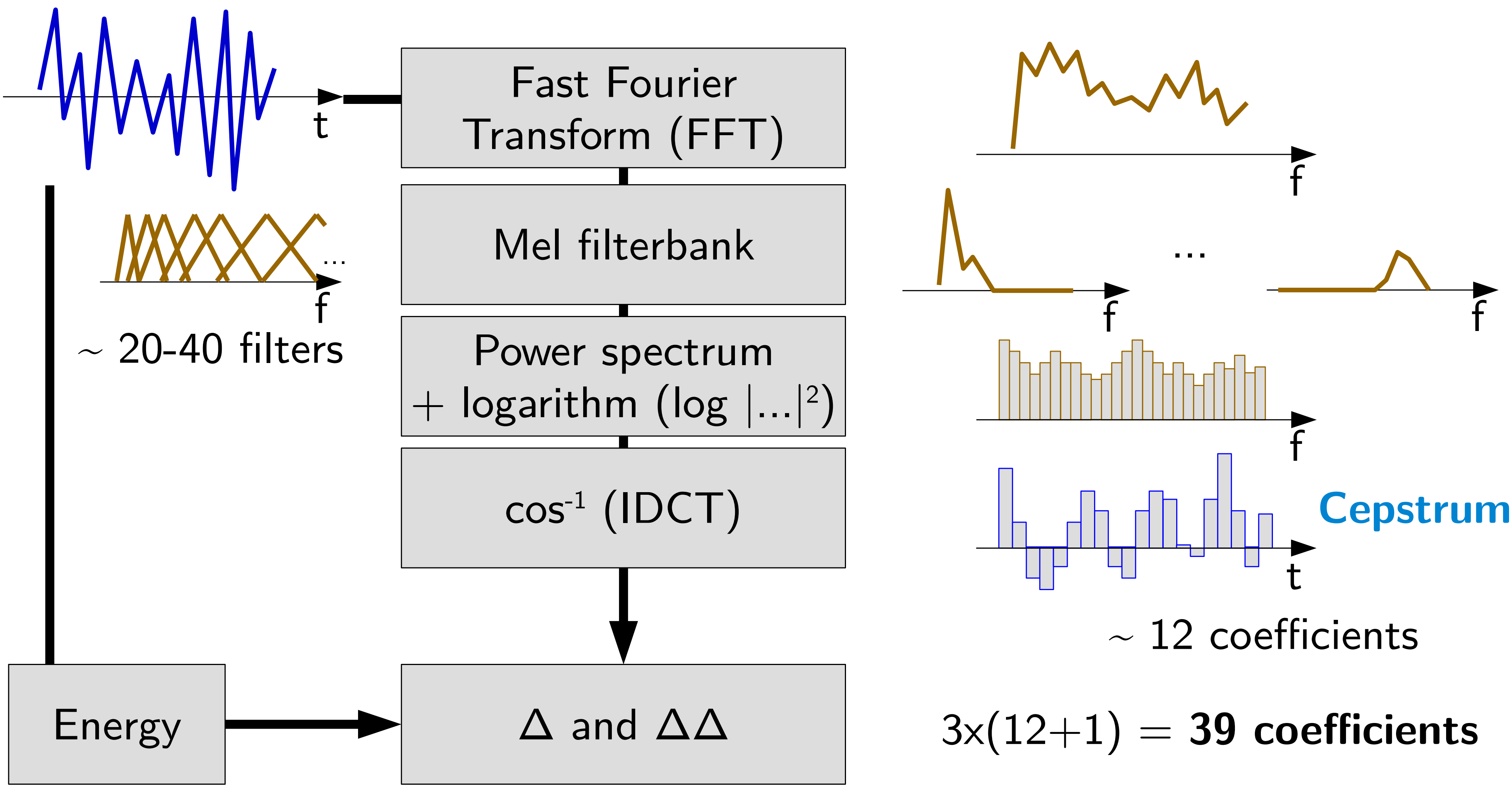


Frame i – 1 Frame i Frame i+1

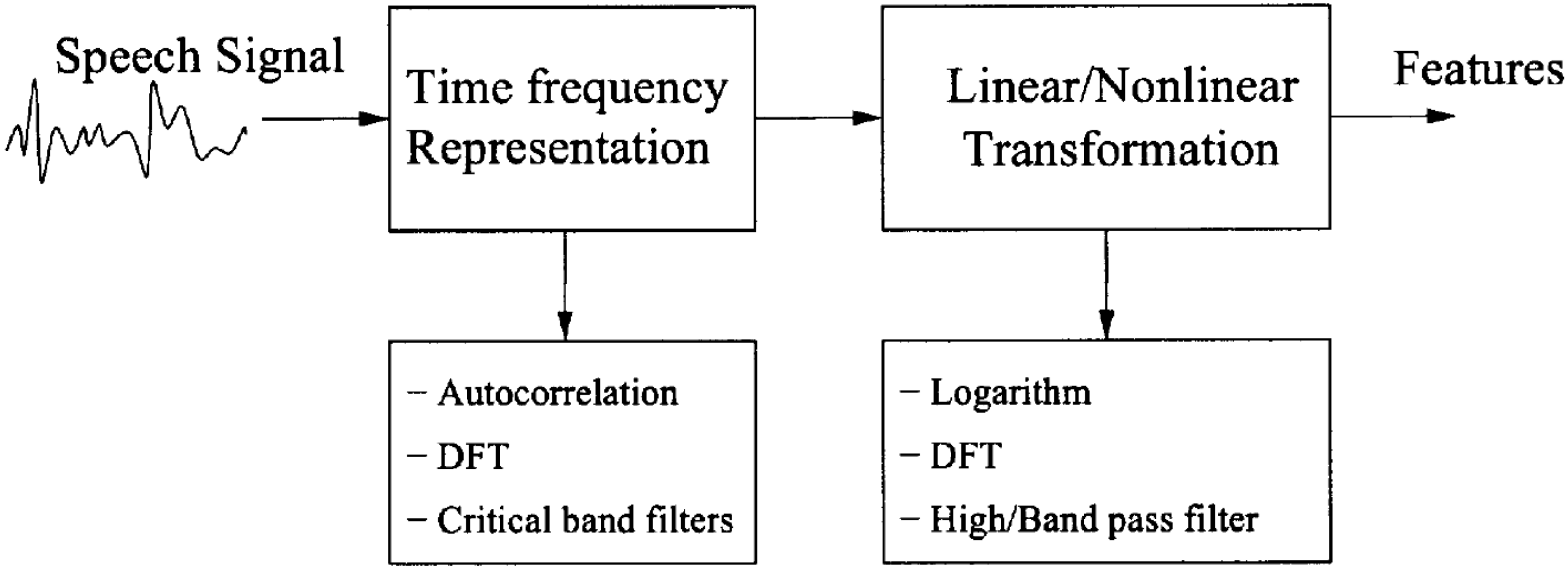
Feature extraction

- Features $=$ Energy $+$ frequencies
- Desirable properties
  - Robust to F0 changes (and F0 harmonics)
  - Robust across speakers
  - Robust against noise and channel distorsion
  - Lowest dimension as possible at equal accurary
  - Non redondancy among features

Mel-Frequency Cepstral Coefficients (MFCC)



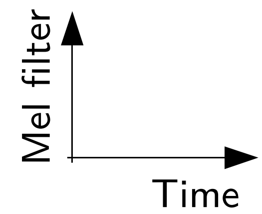
Feature extraction (cont.)

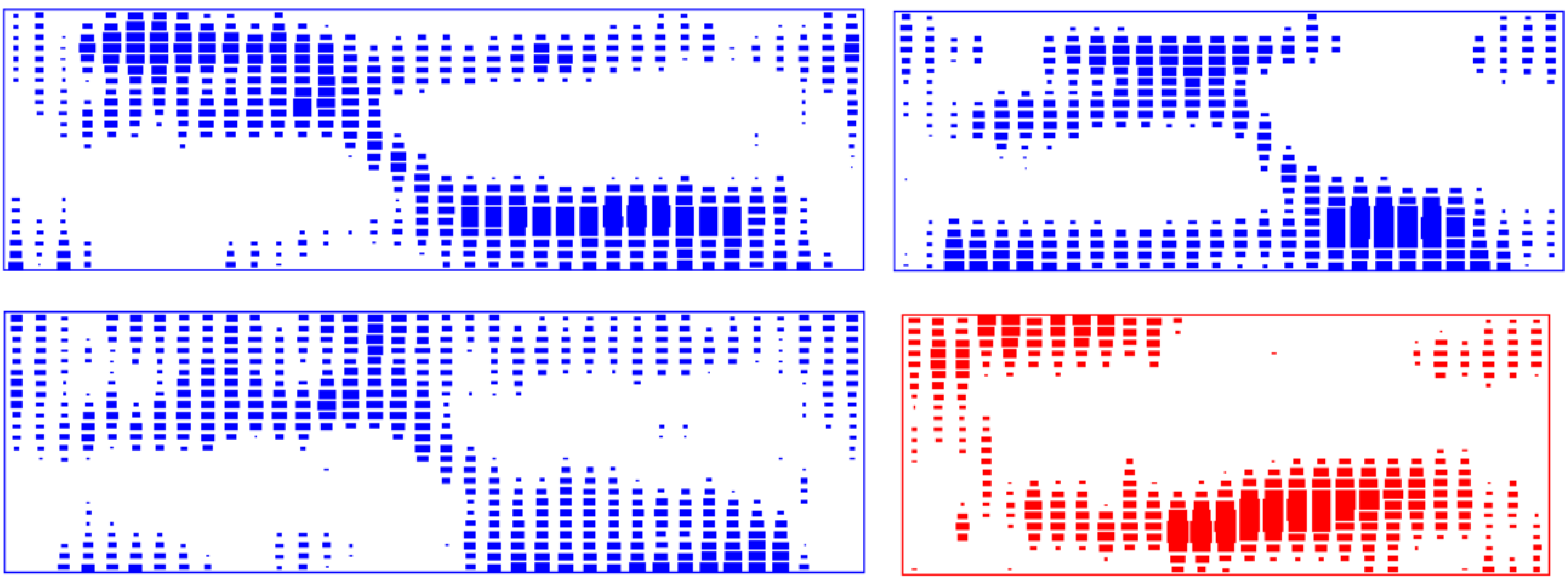


- Other features
  - Perceptual Linear Prediction (PLP)
    - Autoregressive
  - Tandem
    - Discriminative
- Normalization : avoid mismatches across samples
  - Mean/variance normalization

Examples

"zéro" (0 in French)





"trois" (3 in French)

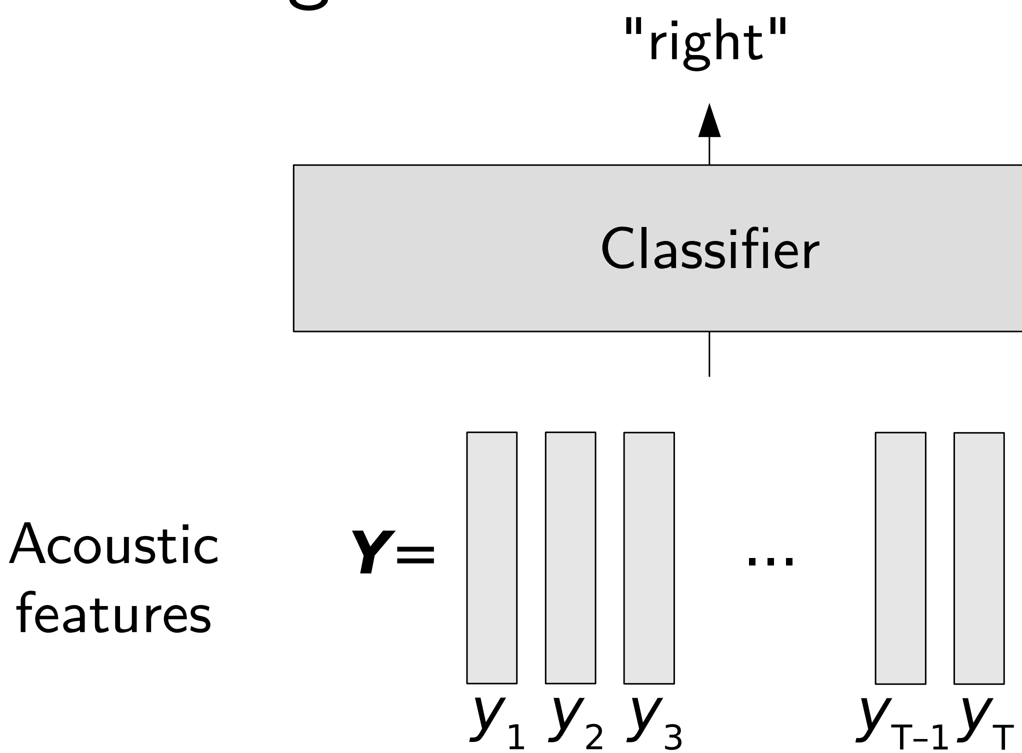
Acoustic modeling

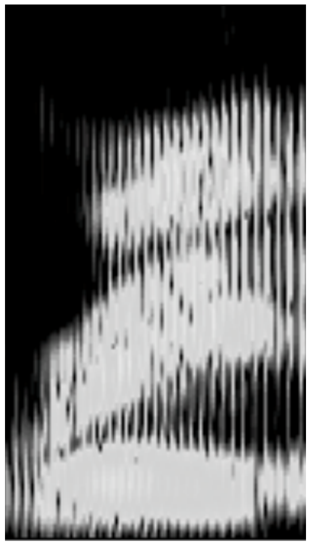
Reading:

- Gales and Young (2007). “The Application of Hidden Markov Models in Speech Recognition”, Foundations and Trends in Signal Processing, 1 (3), 195–304.
- Rabiner and Juang (1989). “An introduction to hidden Markov models”, IEEE ASSP Magazine
- Hinton et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82-97.
- Palaz, D. (2016). Towards End-to-End Speech Recognition

Isolated words

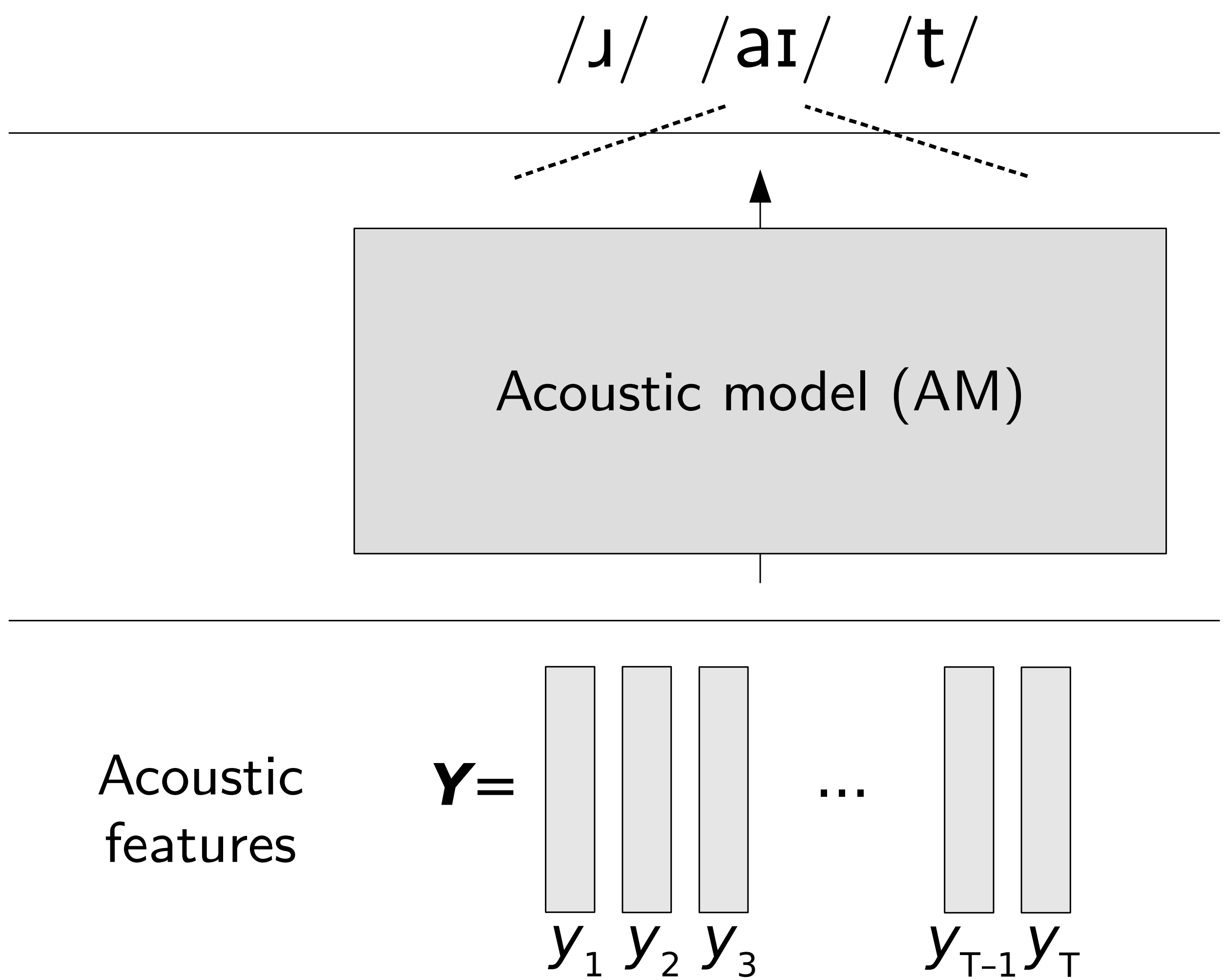
- Pattern recognition

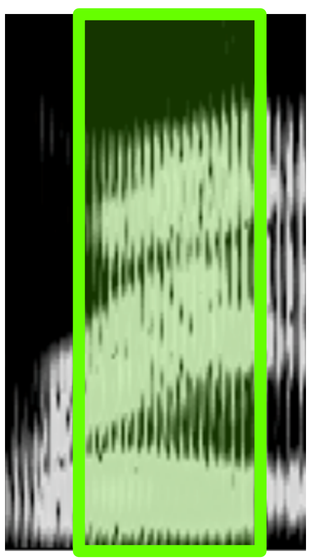




Overview

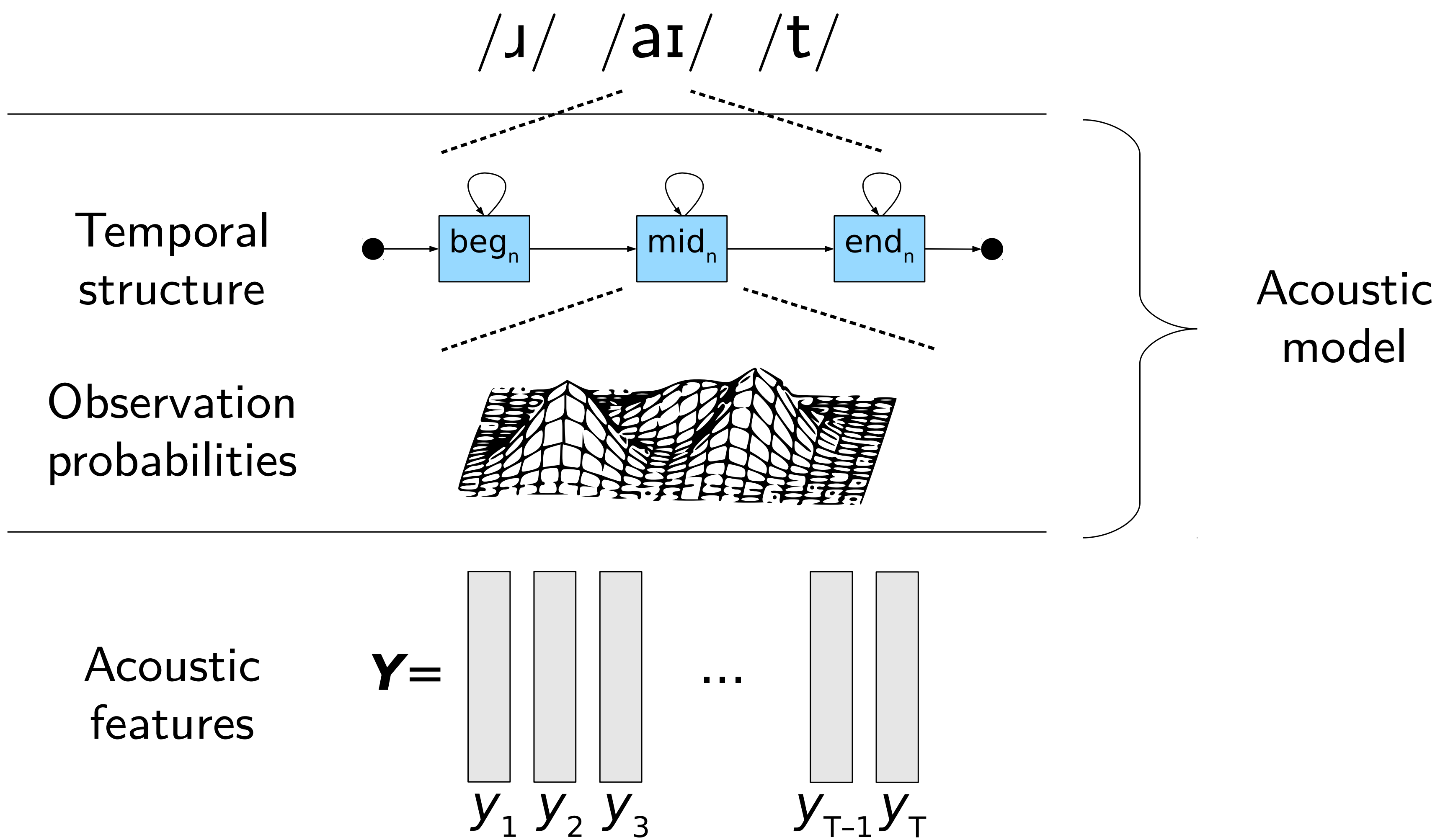
- Decomposition into phonemes





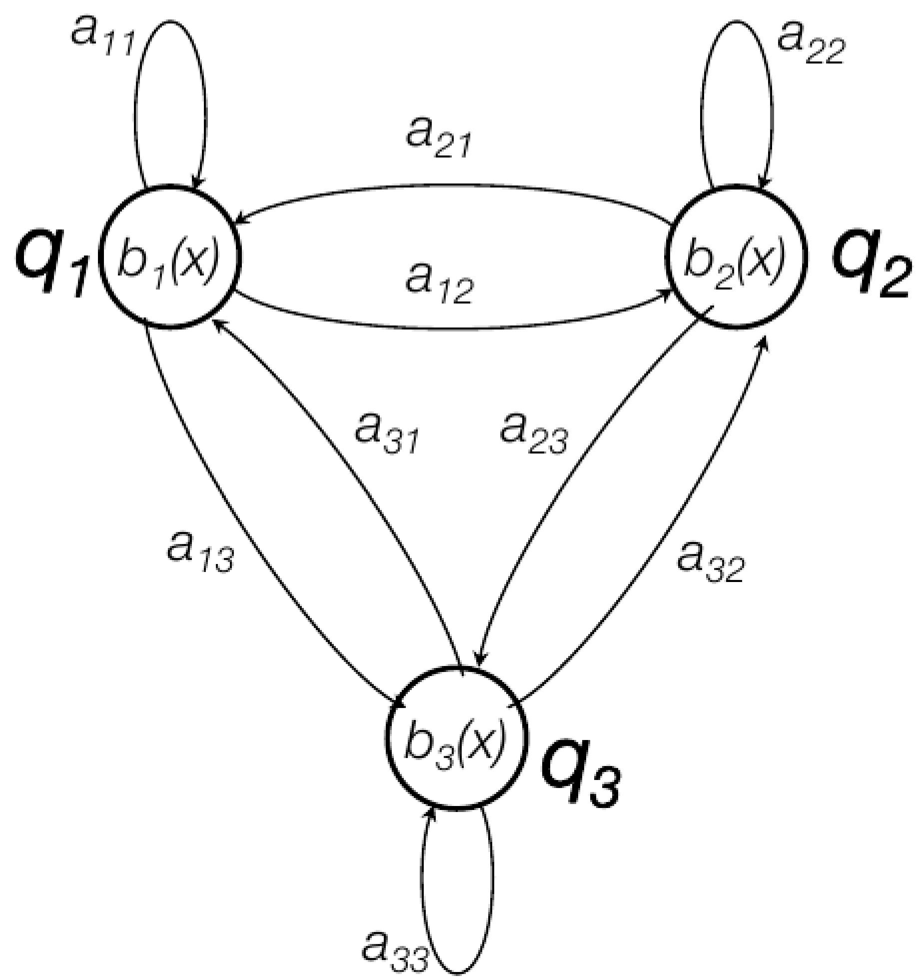
Hidden Markov Models (HMM)

- Overview



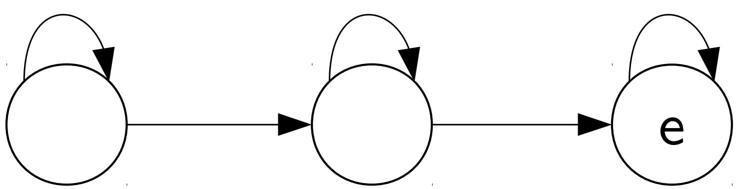
HMMs

- Probabilistic automaton
  - States q_i
    - State at step t: s_t
    - Inititial probability $\pi_i = \Pr(\ s_0 = q_i)$
    - Transition probabilities $a_{ij} = \Pr(s_{t+1} = q_j \mid s_t = q_i)$
  - Outputs
    - Observation at step t: o_t
    - Alphabet of symbols $\mathcal{X} = (x_k)$
    - Emission probability $b_i(x) = \Pr(o_t = x \mid s_t = q_i)$



HMM

- 1 phoneme
  - 3 (or 5-)-state linear HMM : beginning, middle, end
  - Observation independence
    - $\Pr(o_t = x \mid s_t = q_i, s_{t-1} = q_{i'}, \ldots, s_0 = q_{i''})$
      $= \Pr(o_t = x \mid s_t = q_i)$
  - Probability to reach the end state e at frame y_t

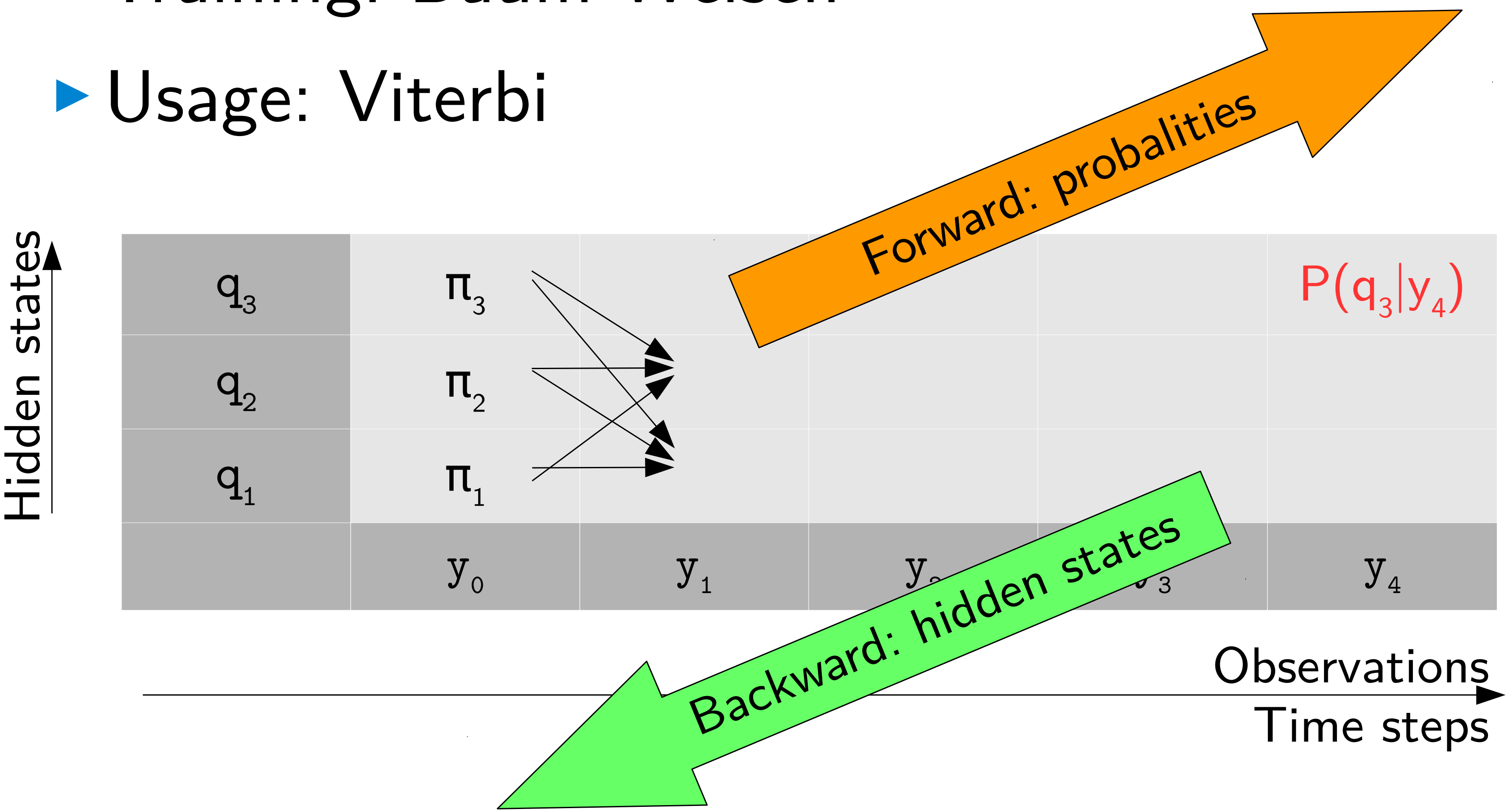


$$P(/a/|t) = P(s_t = e)P(b_e = y_t)$$

- Context-dependent phoneme = triphones
  - Same linear HMMs
  - E.g., /bab/, /bal/, /bap/, etc.
  - State-tying: gather similar HMM state to overcome data sparsity

HMMs

- Training: Baum-Welsch
- Usage: Viterbi



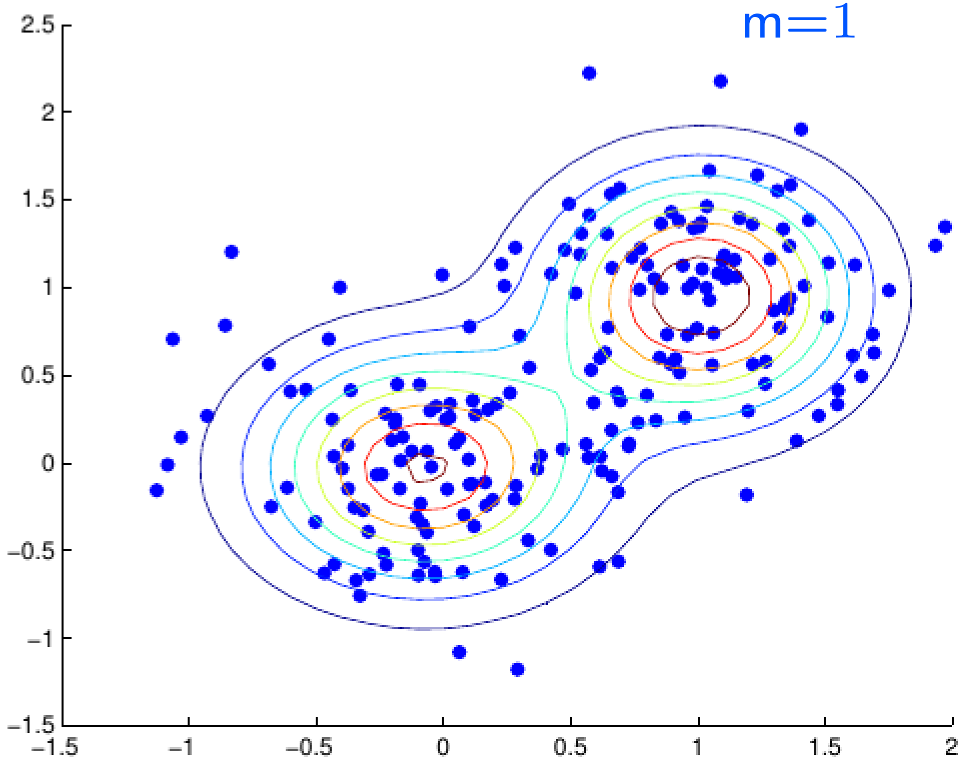
HMM

- Usage:
  - Not finding the hidden state sequence
  - Give probability of each end of phoneme at time t

  $\Rightarrow$ All (context-dependent) phonemes HMMs in parallel
- AM performance
  - Accuracy to recognize the proper phonemes

GMM/HMM

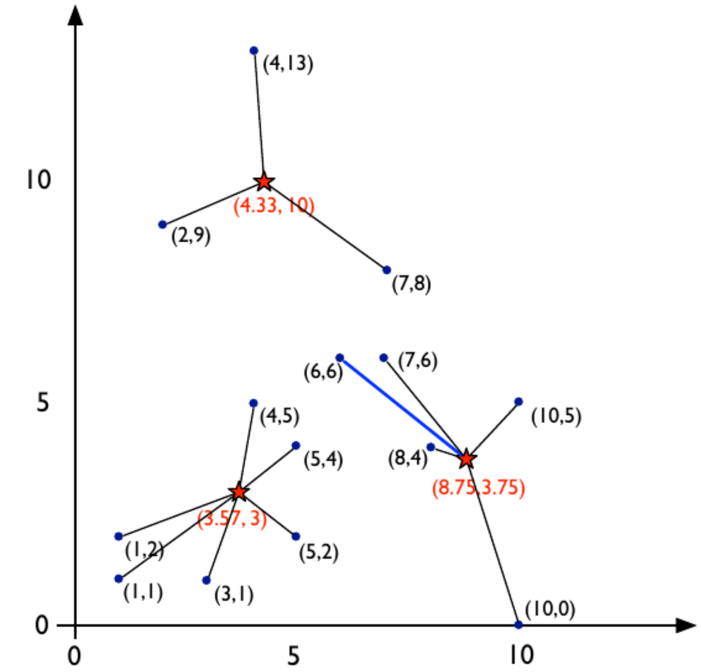
- Gaussian Mixture Model (GMM)
  - M components
  - $\Pr(o_t = \mathbf{y} \mid s_t = q_i) = b_i(\mathbf{y}) = \sum_{m=1}^{M} \mathcal{N}(\mathbf{y}, \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm})$



GMM/HMM

- GMM training
  - M components: K-means
  - PDF estimation
    - $(M+M^2) \times \dim(Y)$ parameters
    - EM algorithm
    - Maximize $\displaystyle P(m|\mathbf{y}) = \frac{p(\mathbf{y}|m)P(m)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|m)P(m)}{\sum_{m'=1}^{M} p(\mathbf{y}|m')P(m')}$
    - Constraint $\displaystyle \sum_{m=1}^{M} P(m|\mathbf{y}) = 1$

    $\rightarrow$ See ADM course

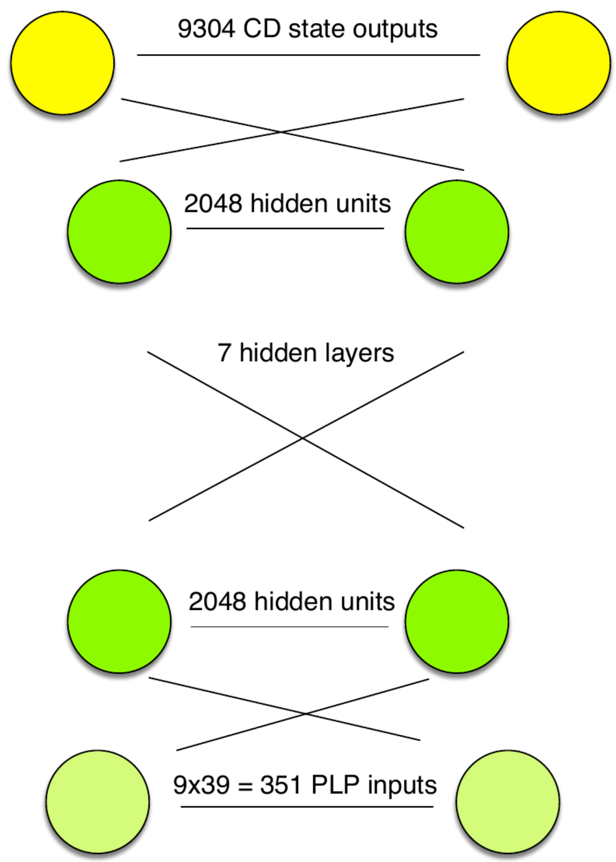


DNN/HMM

- DNN (DNN/HMM)
  - Train an GMM/HMM, label sequence elements
  - Train DNN (supervised learning)

$$\begin{array}{cccl} \mathcal{X} & \times & Q & \rightarrow [0, 1] \\ x & , & q & \rightarrow \Pr(o_t = x \mid s_t = q) \end{array}$$

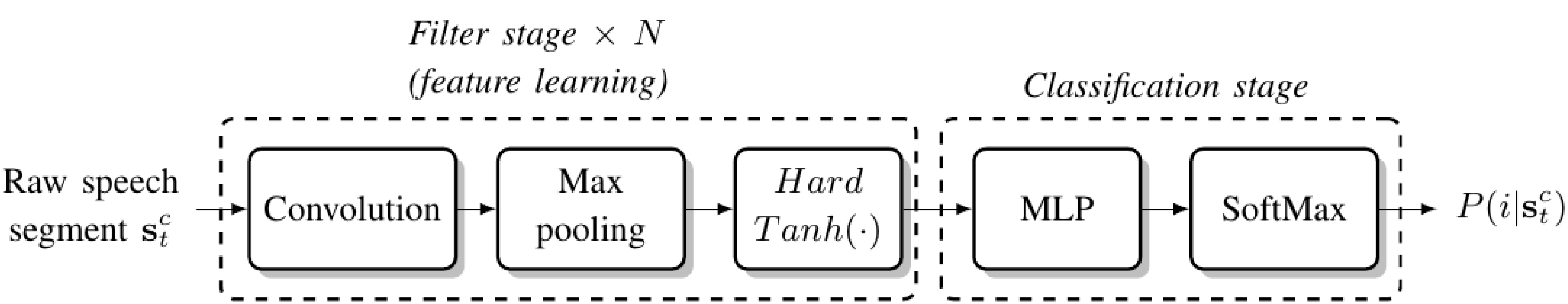
- Feedforward NNs, convolutional NNs
  - Features of step t
    $\mathbf{y}_t \rightarrow$ Phoneme class p
  - With neighbours
    $(\mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{y}_{t+1}) \rightarrow p$



(source: Hinton et al., 2012)

Recurrent NNs, end-to-end models

- Recurrent NNs, LTSMs, etc.
  - (Segment of) Sequence
    $Y_t^C = (y_i)_{t\text{-}C..t} \rightarrow$ sequence of $p_i \rightarrow$ last phoneme p_t
- End-to-end
  - Combine feature extraction and phoneme prediction



(source: Pallaz et al., 2016)

Pronunciation

Reading:

- Jurafsky and Martin (2008). Speech and Language Processing (2nd ed.)
- Rasipuram (2014). Grapheme-based automatic speech recognition using probabilistic lexical modeling.
- Collobert et al. (2016). Wav2Letter: an End-to-End ConvNet-based Speech Recognition System. ArXiv.

Back to ASR

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \mathsf{p}(\mathbf{Y}|\mathbf{W})\ \mathsf{P}(\mathbf{W})$$

- Words are sequences of states **Q**

$$\begin{aligned}
\mathbf{W}^* &= \arg\max_{\mathbf{W}} \mathsf{p}(\mathbf{Y}|\mathbf{Q},\mathbf{W})\ \mathsf{P}(\mathbf{Q},\mathbf{W}) \\
&\approx \arg\max_{\mathbf{W}} \mathsf{P}(\mathbf{Y}|\mathbf{Q})\ \sum_{\mathbf{Q}} \mathsf{P}(\mathbf{Q}|\mathbf{W})\ \mathsf{P}(\mathbf{W}) \\
&\approx \arg\max_{\mathbf{W}} \max_{\mathbf{Q}} \mathsf{P}(\mathbf{Y}|\mathbf{Q}) \underbrace{\boxed{\mathsf{P}(\mathbf{Q}|\mathbf{W})}}_{\text{Pronunciation model}} \mathsf{P}(\mathbf{W})
\end{aligned}$$

Pronunciation dictionary / Lexicon

- Most basic way
- Word $\rightarrow$ phoneme sequence
- No probabilities
- Written by human experts

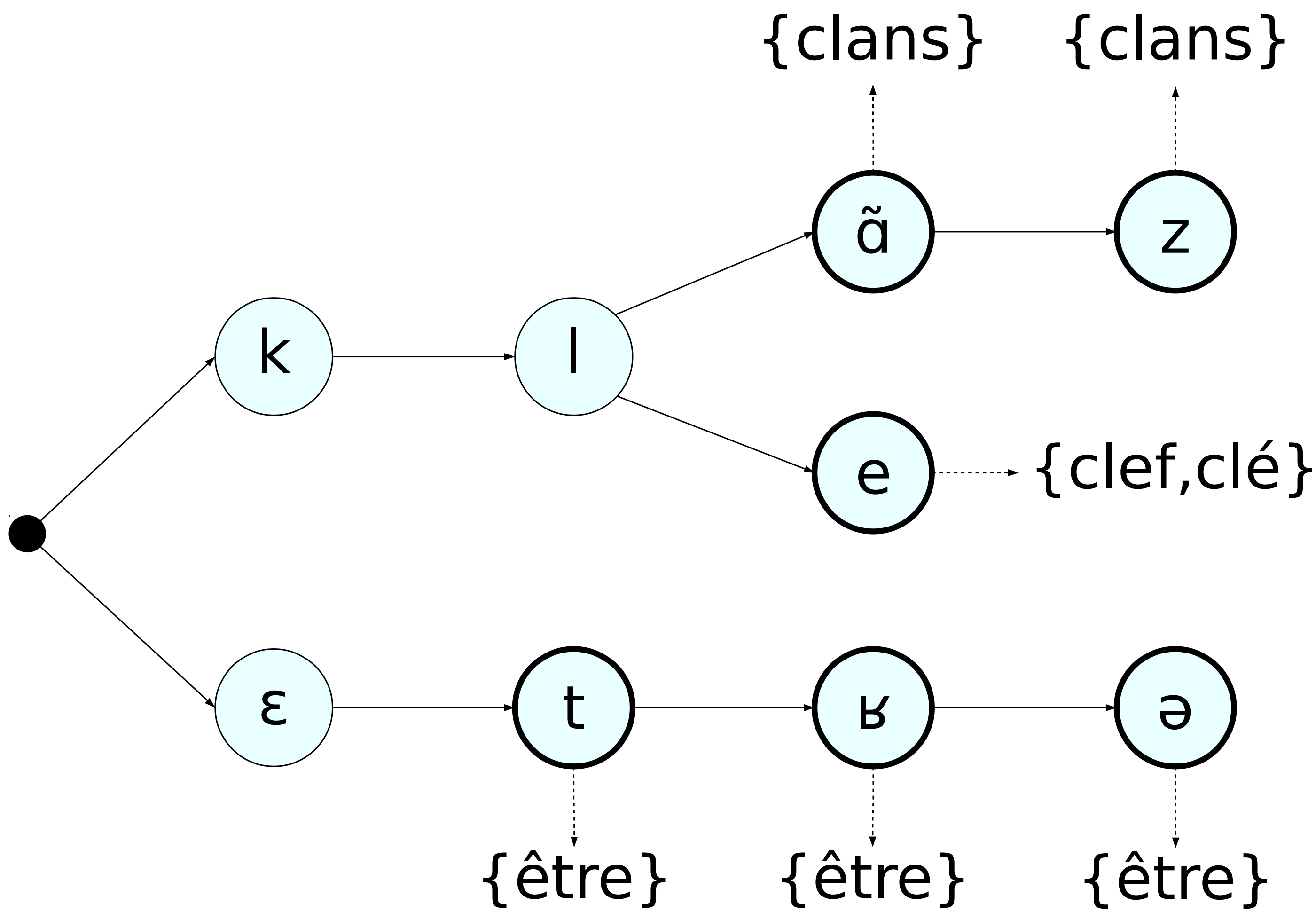
  $\rightarrow$ Key aspect in ASR accuracy
- Coverage
  - AM training set: all words to train HMMs (or whatever)
  - Maximize coverage over some representative texts

N-M relation

- English
  - "hello" → /hɛloʊ/
  - "hello" → /həloʊ/
  - "there" → /ðɛɹ/
  - "their" → /ðɛɹ/
- Français
  - "les" → /lɛ/
  - "les" → /le/
  - "les" → /lɛz/
  - "les" → /lez/
  - "clans" → /klɑ̃/
  - "clans" → /klɑ̃z/
  - "clé" → /kle/
  - "clef" → /kle/
  - "être" → /ɛtʁə/
  - "être" → /ɛtʁ/
  - "être" → /ɛt/

Lexical tree

- Prefix factorization



Out Of Vocabulary (OOV) words

- Constructing a dictionary involves
  - 1 Selection of the words in the dictionary—want to ensure high coverage of words in test data
  - 2 Representation of the pronunciation(s) of each word
- OOV rate: percent of word tokens in test data that are **not** in the ASR system dictionary
- OOV rate increase => WER increase
  - 1,5-2 errors per OOV word (>1 because loss of context)

Vocabulary content

- Words
- Multi-words: frequent sequences of words
  - "want to" $\rightarrow$ "want_to"
  - "je suis" $\rightarrow$ "je_suis"

  $\rightarrow$ Handling of pronunciation variants
- Subword units (morphemes, characters, etc.)
  - OOV words
  - Character-based languages
  - Agglutinatives languages

Word normalization

- Many variants
  - Hong-Kong, Hong Kong
  - U.N., UN, U. N.
  - Trinity College, new college
  - 2, two
  - Mr, Mister
  - 100m, 100 meters
- Automatic learning/discovery based on knowledge resources (Wikipedia, Wiktionary, WordNet, etc.)

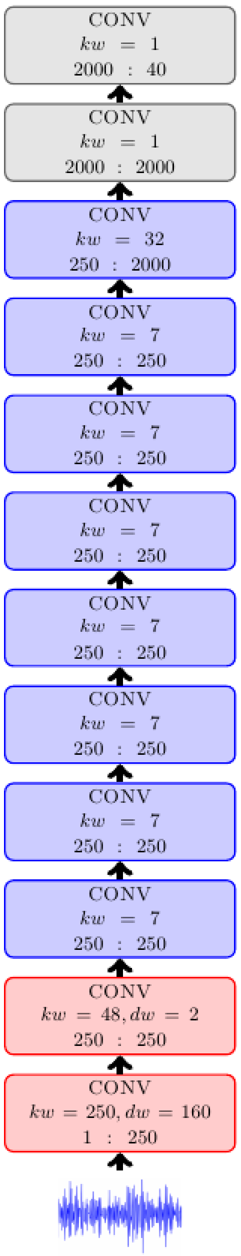
Current topics

- Pronunciation variants or alternative pronunciations
  - Grapheme-to-phoneme (G2P) models: automatic learning of pronunciations of new words
  - Probability distribution over possible pronunciations
- Cobebook learning : joint learning of the inventory of subword units and the pronunciation lexicon

  $\rightarrow$ Minimum description length (MDL)
- Sub-phonetic / articulatory feature model

Current topics

- Grapheme-based acoustic modelling (Rasipuram, 2014)
  - Character level
  - No more pronunciation modelling entirely
- Grapheme-based speech recognition: wav2letter (Collovert et al., 2016)
  - End-to-end approach



Language modeling

Reading:

- Jurafsky and Martin (2008). Speech and Language Processing (2nd ed.)
- Bengio et al. (2006), “Neural probabilistic language models” (sections 6.1, 6.2, 6.3, 6.6, 6.7, 6.8), Studies in Fuzziness and Soft Computing Volume 194, Springer, chapter 6.
- Mikolov et al (2011), “Extensions of recurrent neural network language model”, Proc. of ICASSP.
- R Jozefowicz et al (2016), “Exploring the Limits of Language Modeling”. ArXiv.

Constraints

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \mathsf{p}(\mathbf{Y}|\mathbf{W})\ \mathsf{P}(\mathbf{W})$$

- What the speaker is allowed to say
- Constrained grammar
- Binary decision
- Task-oriented
- $+$ More precise
- – Less flexible

- What the speaker may say
- Free grammar
  - Given the vocabulary
- Probabilities over possible sequences
  - A priori: trained on some text

Regular/Context-free grammar

<Root> = <Date>

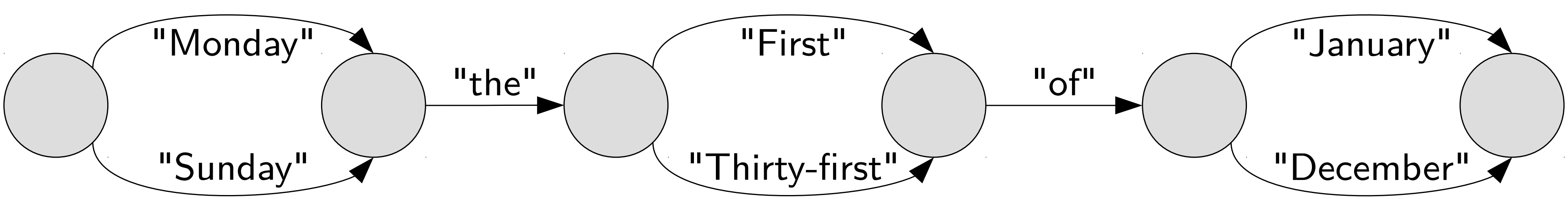
<Date> = <Day> "the" <Ith> "of" <Month>

<Day> = "Monday" | "Tuesday" | ... | "Sunday"

<Ith> = "first" | "second" | ... | "thirty-first"

<Month> = "January" | ... | "December"

- No training data to be collected
- Finite state automaton/Pushdown automaton



- Grammar can be made probabilistic

Statical language modeling

- Idea
  - Cover all possible sequence (V* = V x V x V x ...)
  - Disambiguate acoustically ambiguous sequences
    "recognize speech", "wreck a nice beach"
- Sequence $\mathbf{W} = w_1 \ldots w_N$
- Chain rule $P(w_1, w_2, \ldots w_N) = P(w_1) \prod_{i=2}^{N} P(w_i | \overbrace{w_1 \ldots w_{i-1}}^{\text{History } h})$
- Maximum likelihood estimation (MLE)

$$P(w_i | w_1 \ldots w_{i-1}) = \frac{C(w_1 \ldots w_i)}{\sum_v C(w_1 \ldots w_{i-1} v)}$$

, with $C(w_1 \ldots w_i)$ observed count in the training data

Smoothing and Backoff

- What if never observed during training?

  $\rightarrow$ The longer sequence, the more zero-counts

- History of words

$$P(w_1, w_2, ...w_N) \approx P(w_1) \times \prod_{i=2}^{N} P(w_i | \Phi(w_1 ... w_{i-1}))$$

- Smoothing
  - Redistribute probability mass from observed to unobserved events : change counts and renormalize
    - Absolute discouting, Kneser-Ney smoothing
- Backoff
  - Link unseen events to the most related seen events

n-gram model

- Truncate the word history

$$P(w_i|h) = P(w_i|w_{\mathbf{i\text{-}n+1}}...w_{i-1})$$

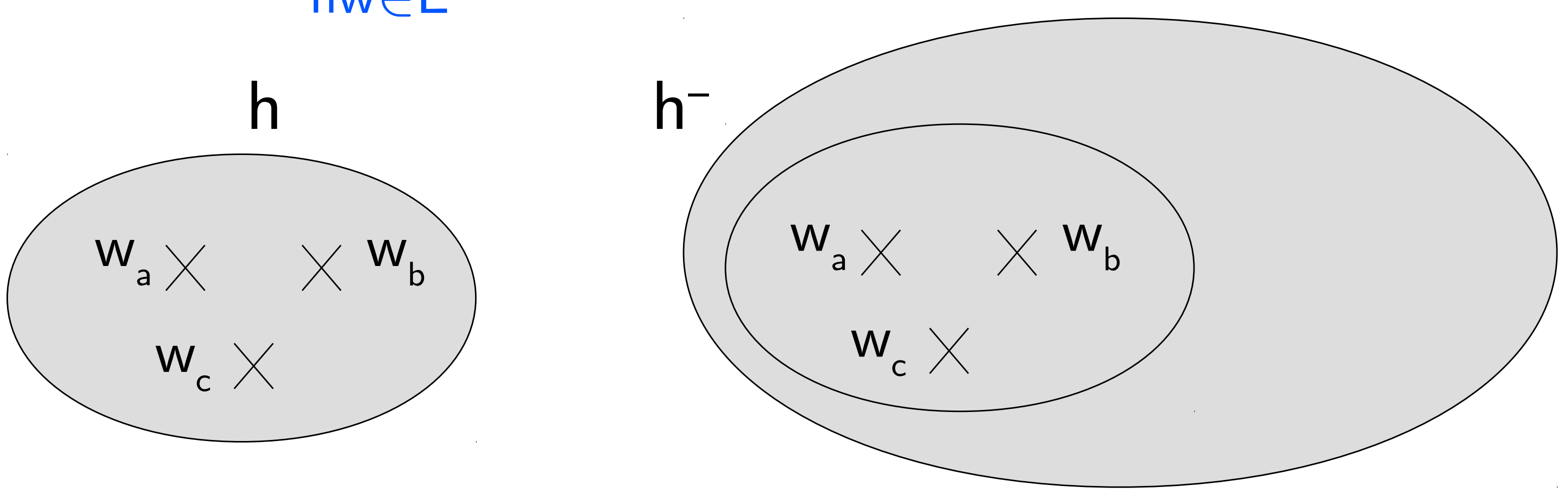
  with n usually 2..5

- $n = 1 \rightarrow$ "unigram", $n = 2 \rightarrow$ "bigram", $n = 3 \rightarrow$ "trigram"
- Backoff
  - Unseen $w_a w_b w_c \rightarrow$ fallback to $w_b w_c$
  - $P(w_c|\underbrace{w_a w_b}_{h}) = P(w_c|\underbrace{w_b}_{h^-})\ \beta(w_a w_b)$

n-gram model

$$P(w|h) = \begin{cases} P^*(w|h) & \text{if } hw \in E \text{ (observed events)} \\ \beta(h) \times P(w|h^-) & \text{otherwise} \end{cases}$$

$$\beta(h) = \frac{1 - \sum_{hw \in E} P^*(w|h)}{1 - \sum_{hw \in E} P^*(w|h^-)}$$

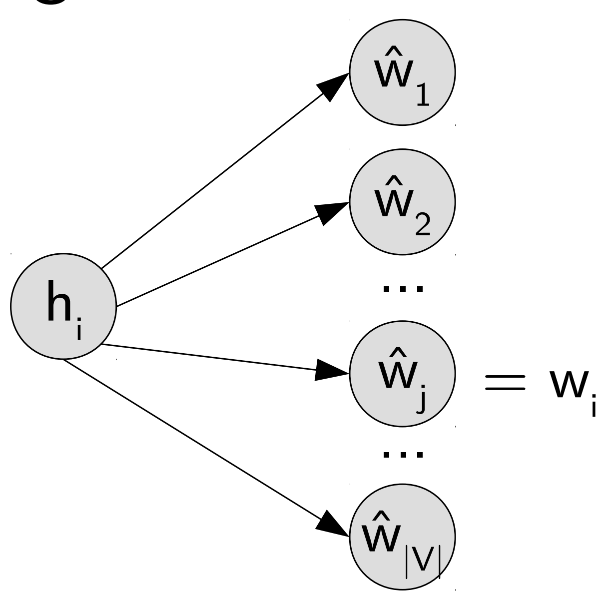


Perplexity

- How well a text T is predicted by a model M?
- Definition 1
  - Cross-entropy $$H(P_T, P_M) = \sum_{w_i \in T} P_T(w_i) \log_2 P_M(w_i|h_i)$$
  - Perplexity $$PPL_M(T) = 2^{-H(P_T, P_M)}$$
- Definition 2
  - Average log-likelihood of M over T (n words) $$L(T|M) = \frac{1}{n} \times \sum_{w_i \in T} \log_2 P_M(w_i|h_i)$$
  - Perplexity $$PPL_M(T) = 2^{-L(T|M)}$$

Perplexity

- The lower, the better
- Best theoretical perplexity $= 1$
- Intepretation
  - Branching factor



$E(j) = PPL$

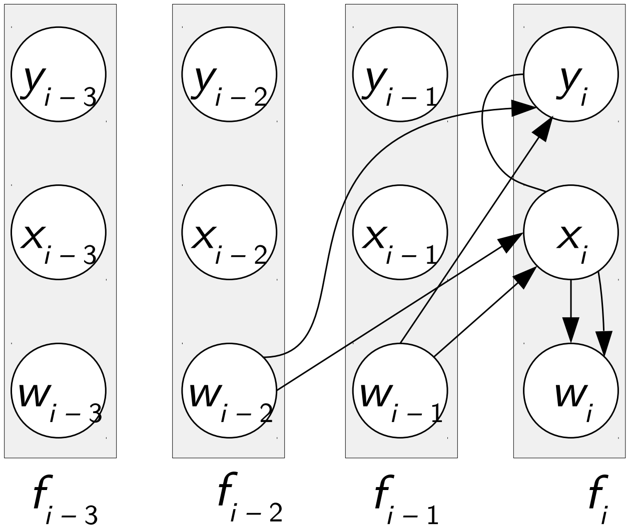
Advanced n-gram models

- Factored language models
  - 1 word $w_i \rightarrow$ 1 feature vector $\mathbf{f}_i = (w_i, x_i, y_i, ...)$

$$P(w_i | w_{\mathbf{i-n+1}} ... w_{i-1}) = P(\mathbf{f}_i | \mathbf{f}_1 \cdots \mathbf{f}_{i-1})$$

  - Data sparsity $=>$ feature dependencies

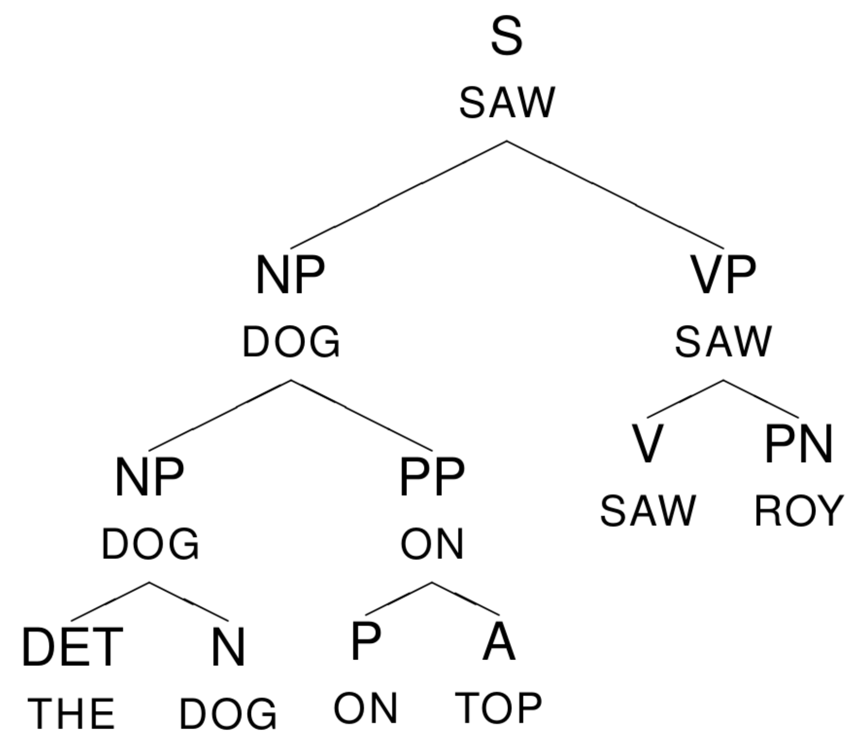
$$\begin{aligned} & P(w_i | x_i, y_i) \\ \times\ & P(x_i | w_{i-2}, w_{i-1}) \\ \times\ & P(y_i | w_{i-2}, w_{i-1}) \end{aligned}$$



  - Backoff scheme

Advanced n-gram models

- Structured language models (LMs)
- Long-span/distant dependencies
- Syntax parsing
  $\rightarrow$ Grammatical function
  $+$ headword
- Idea: condition words on parent's information
- Difficulty: online parsing



Cache/trigger/topic models

- Cache assumption: words said once may be said again
- Trigger assumption: some words may increase the probability of other words later on
- Extension to topic models
- Additional models

Exponential (MaxEnt) models

- Decompose (h, w) into features functions $f_j(h, w)$

$$P(w_i | w_1 ... w_{i-1}) = \frac{1}{Z(h)} \exp\left(\sum_j \lambda_j \times f_j(h, w_i)\right)$$

  - With Z(h), normalization factor
  - λ_j, parameters to be optimized
- Feature function denote a characteristic
  - Have been observed together
  - Is syntactically correct
  - Is thematically correct
  - Etc.

  $\rightarrow$ Binary value

Exponential (MaxEnt) models

- Training
  - Maximum entropy of the model
  - Under constraints for each feature function f_j

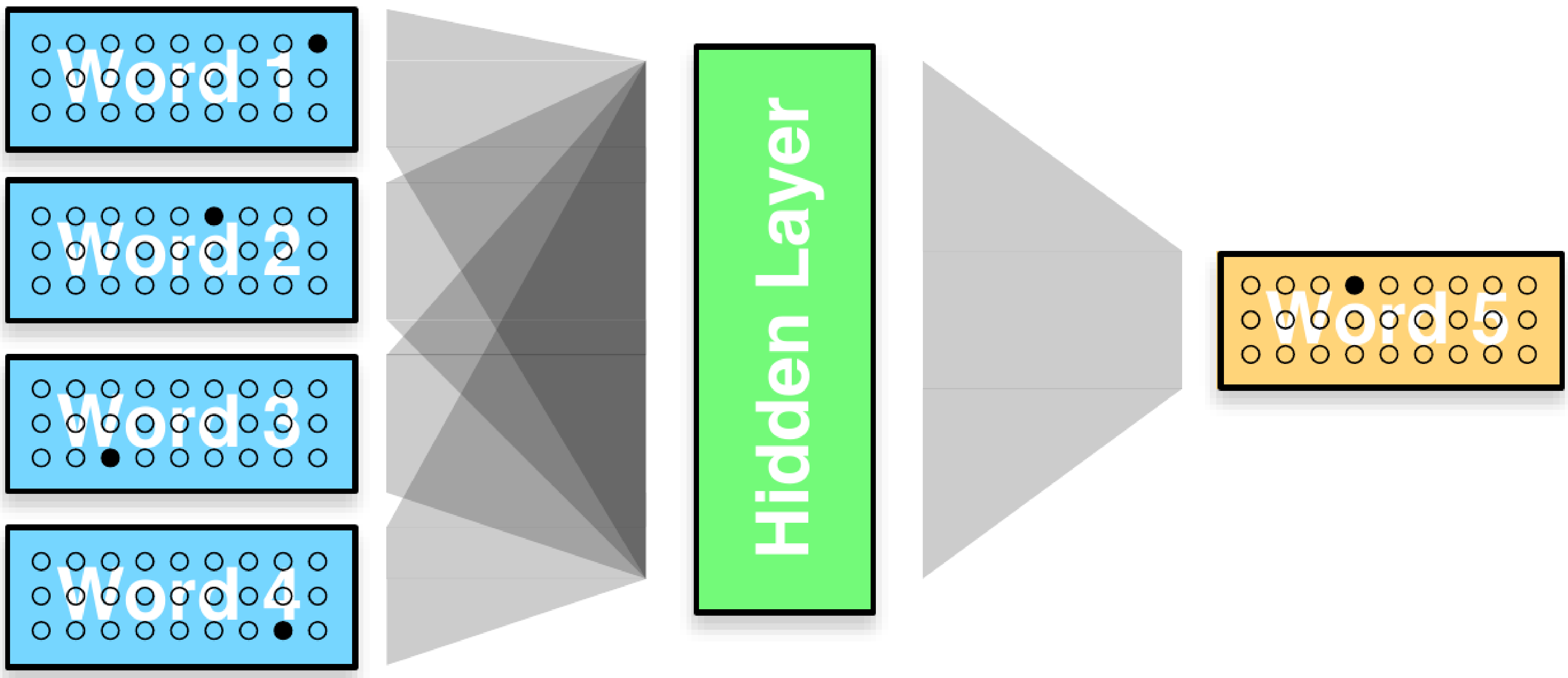
$$\sum_{h,w} P[h,w] \times f_j(h,w) = K_j$$

with K_j, probability mass (usually observed in training data)

- Iterative algorithms
  - Long
  - No smoothing
  - Not always better than n-gram MLE

Neural network LMs

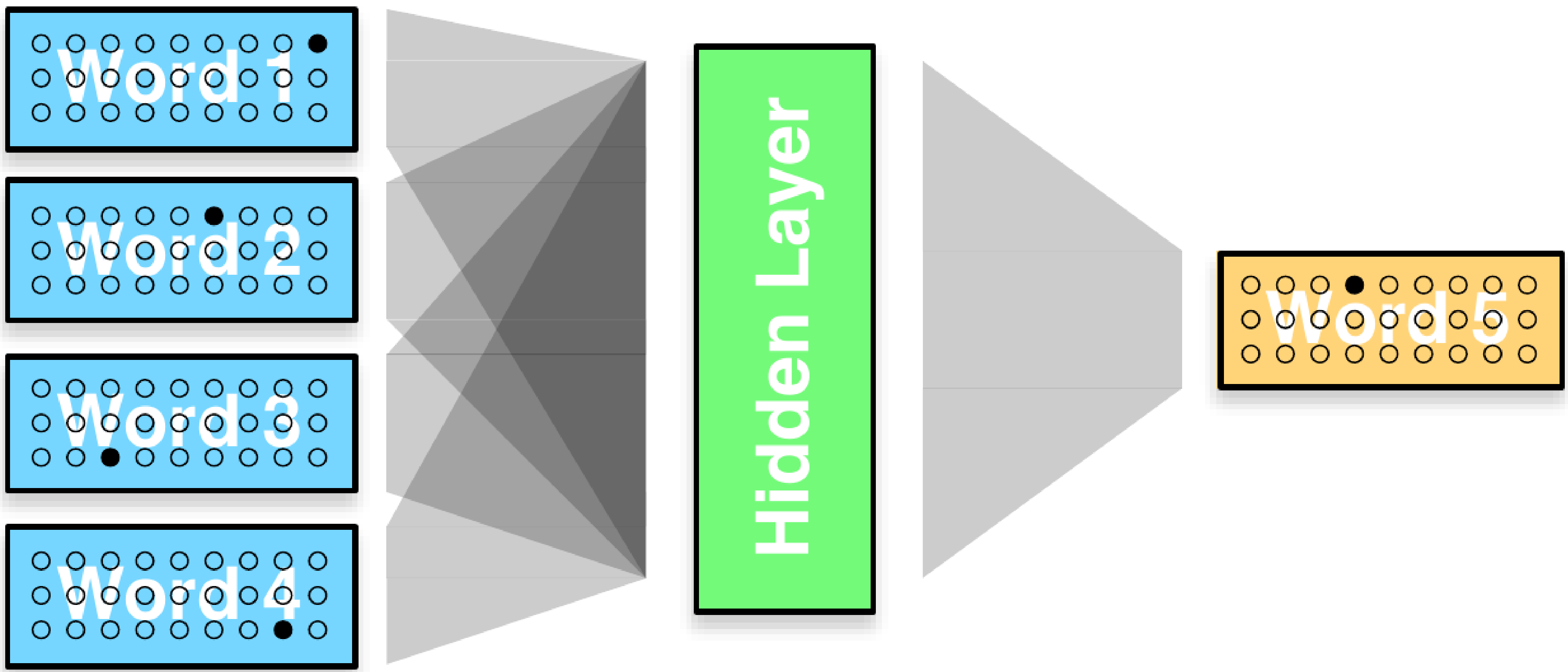
- Feed-forward approach



(source: Koehn, 2016)

Neural network LMs

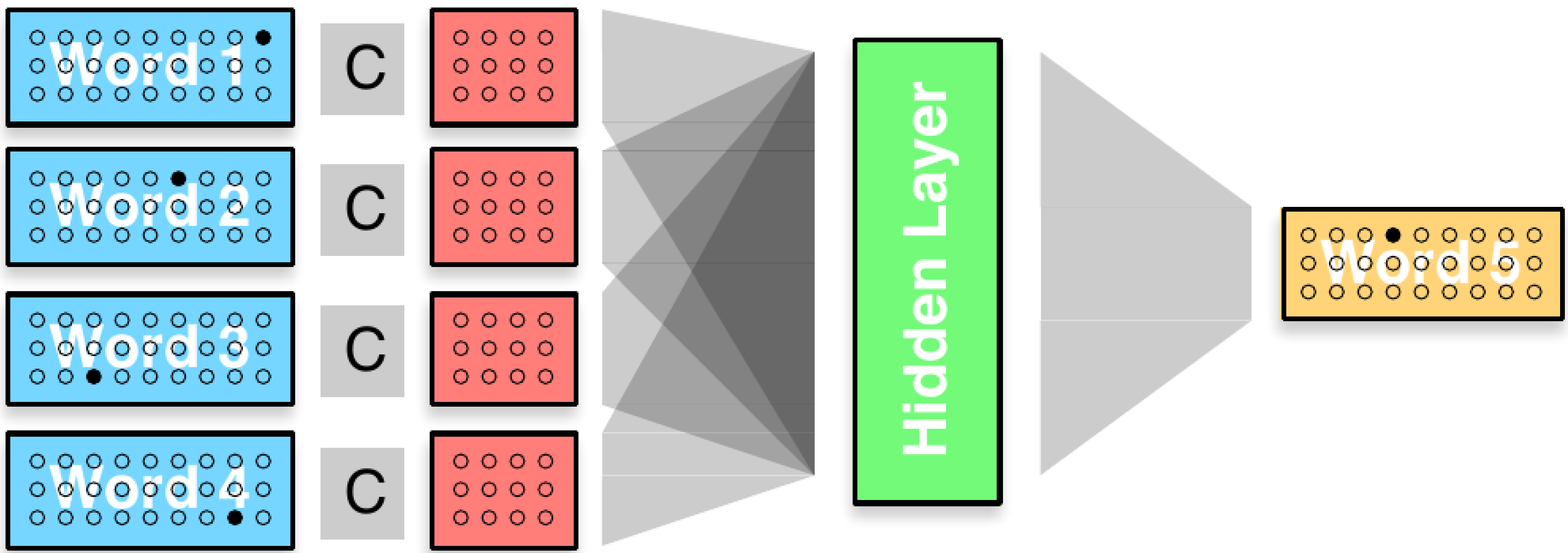
- Feed-forward approach



(source: Koehn, 2016)

Neural network LMs

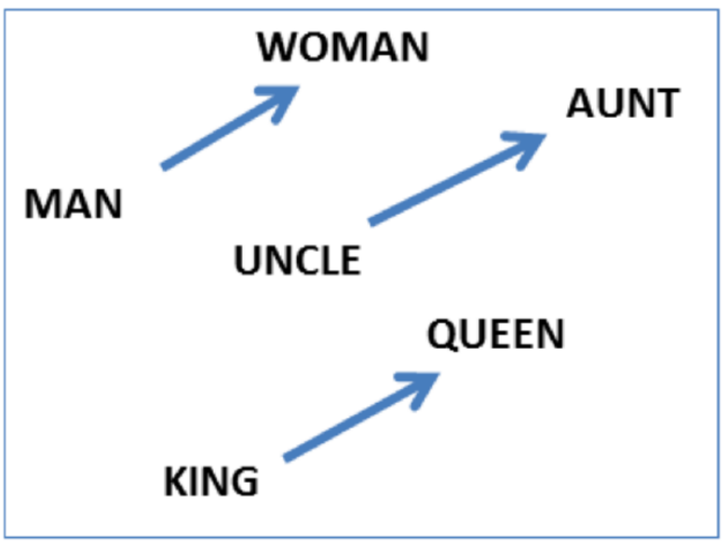
- Word embedding
- Shared weights C

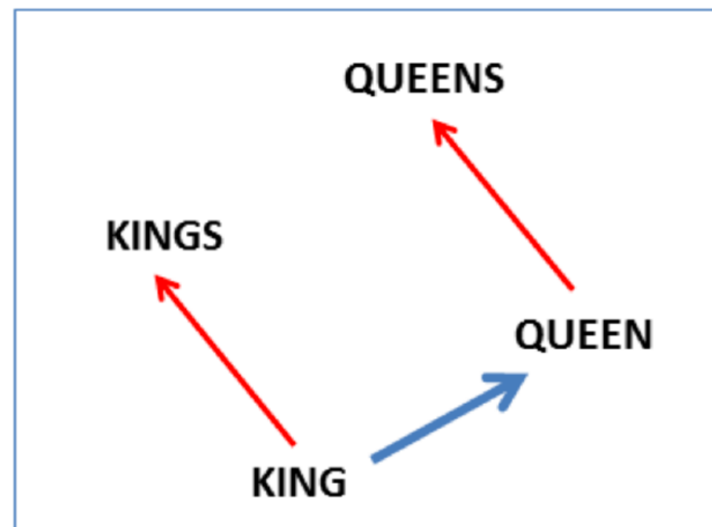


(source: Koehn, 2016)

(Word embeddings)

- Projection into a continuous space $\mathbb{R}^d$
- Topological properties
  - Semantics
  - Morpho-syntax

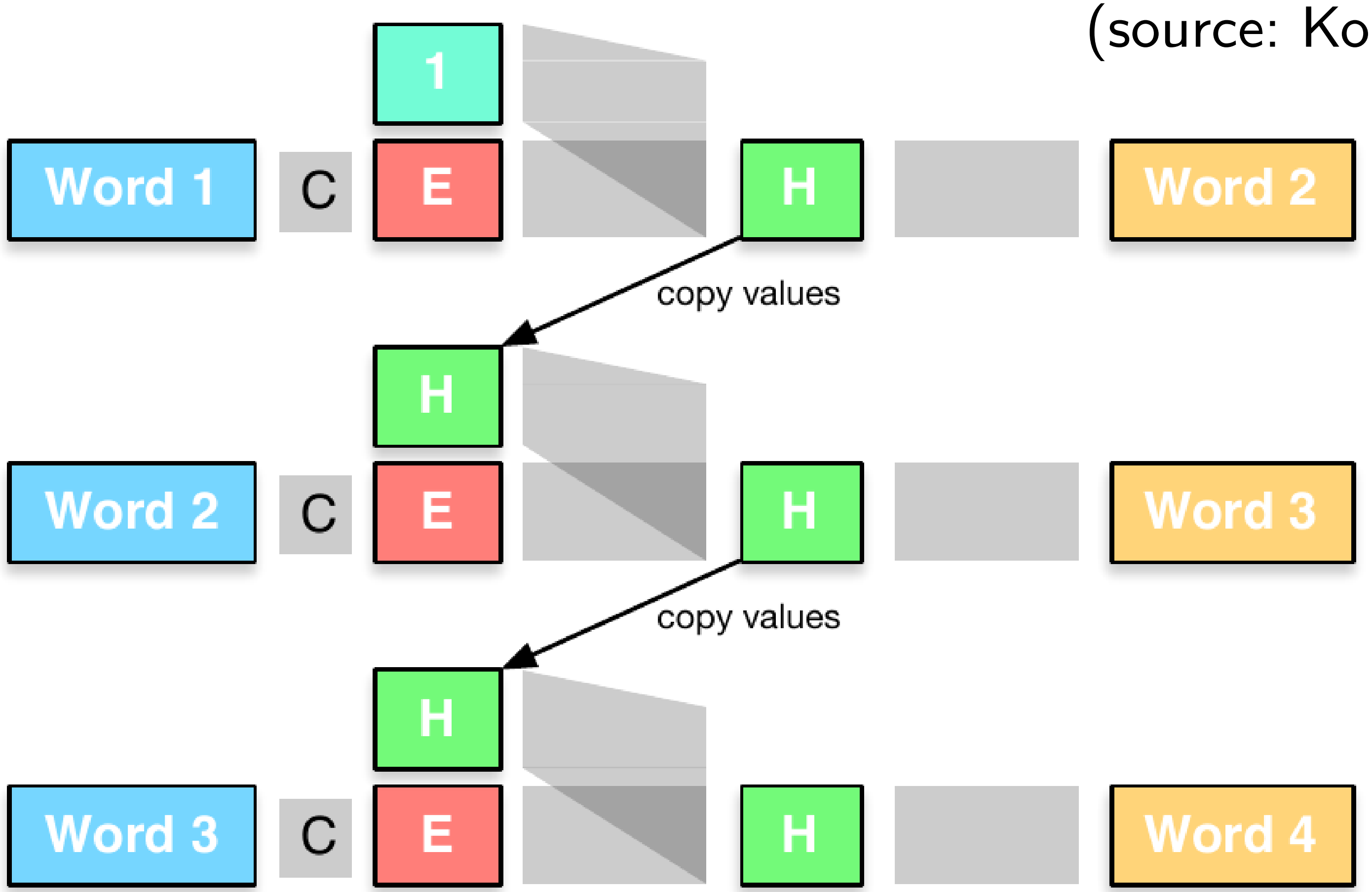




Recurrent neural network LMs

- Build embeddings of word histories

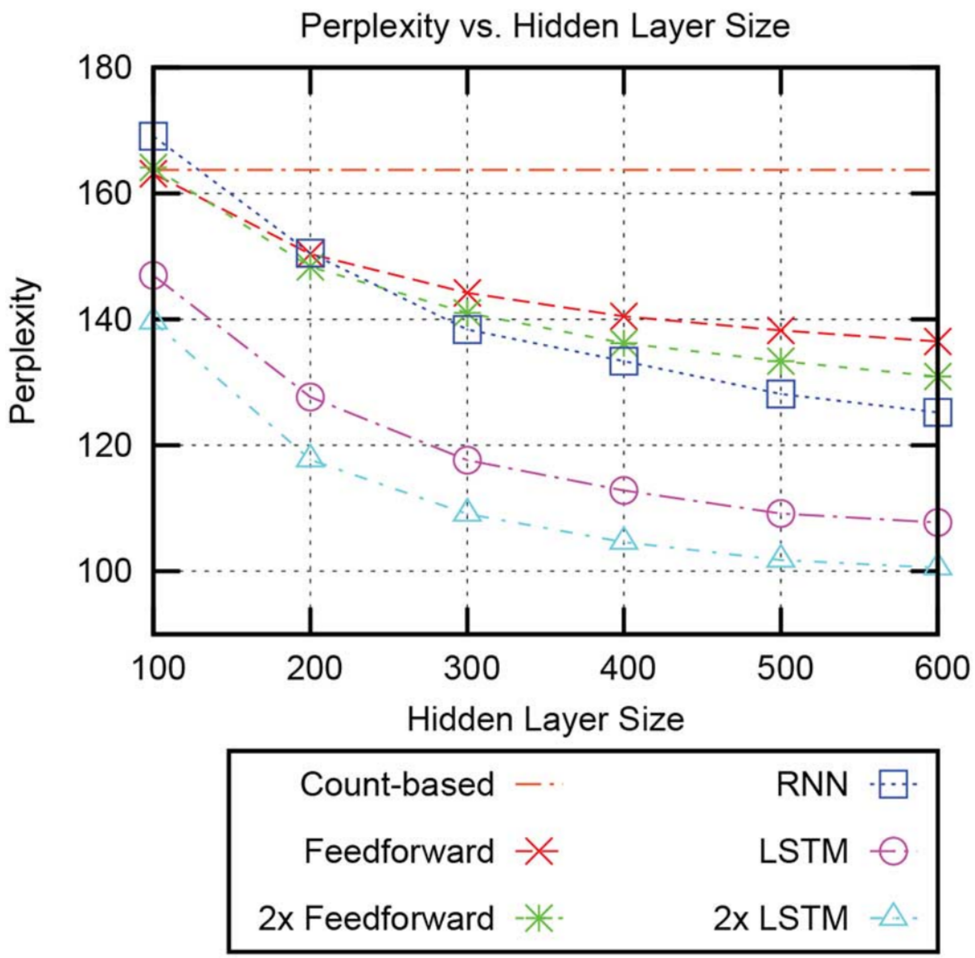
(source: Koehn, 2016)



Long short term memories (LSTM)

- Recurrent neural networks with **forgetting** mechanisms

  ⇒ Important information is remembered longer



(source: Sundermeyer et al., 2015)

Decoding

Reading:

- Ney and Ortmanns (1999). Dynamic programming search for continuous speech recognition. IEEE Signal Processing Magazine, 16(5), 64-83.
- Mohri et al. (2008). “Speech recognition with weighted finite-state transducers.” In Springer Handbook of Speech Processing, pp. 559-584.
- Mangu et al. (2000). Finding consensus in speech recognition: word error minimization and other applications of confusion networks. Computer Speech & Language, 14(4), 373-400.

Beam search decoding

- Need: find

LM scale factor

$$W^* = \arg\max_{W} \underbrace{p(Y|W) \times P(W)^{\psi} \times I^{|W|}}_{score(W)}$$

Word insertion penalty

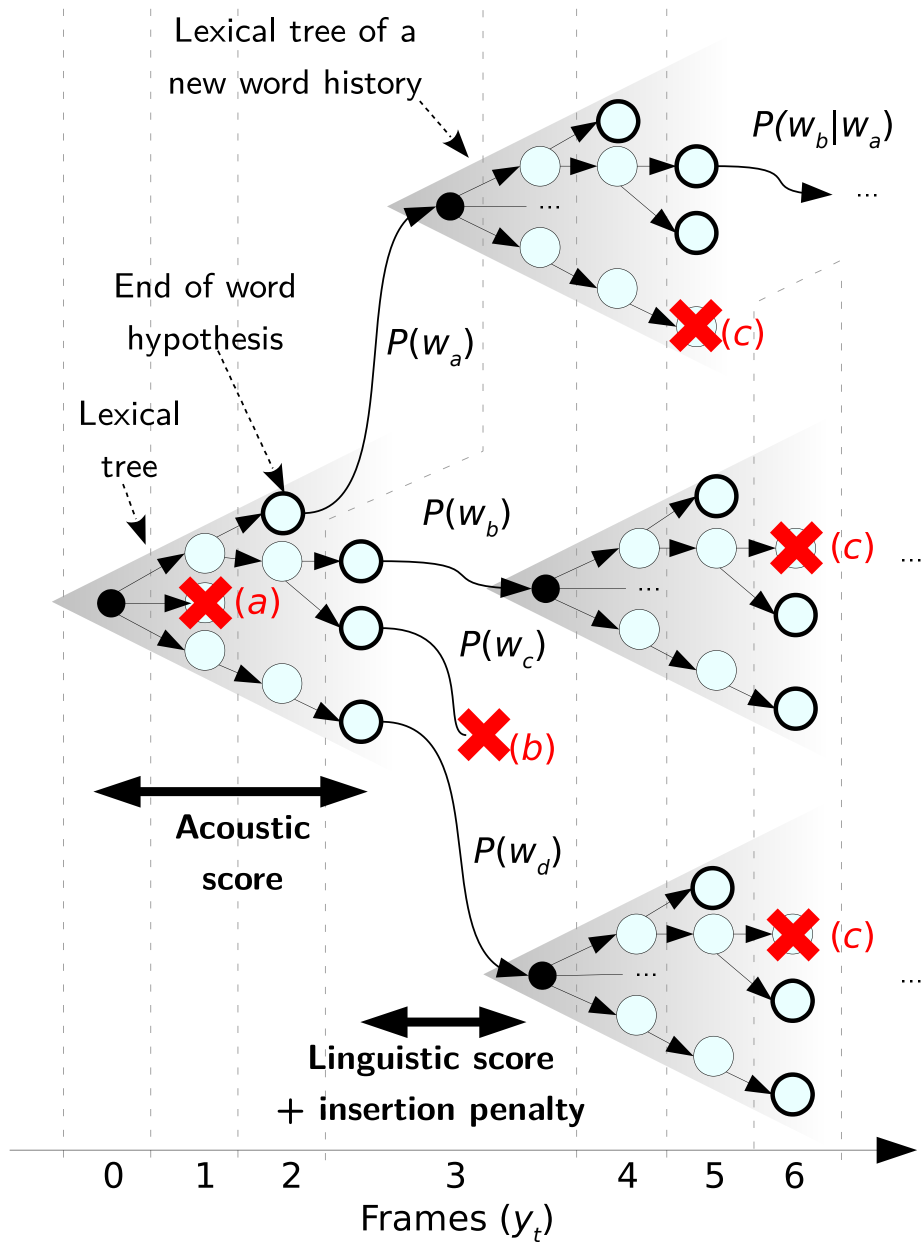
while not exploring the whole search space

- Solution: beam search
  - Frame-synchronous (start at $t = 0$)
  - Idea: Parallel explorations limited a maximum of K active states
  - Advantage: memory and time efficient due to pruning of low interst partial hypotheses

Beam search decoding

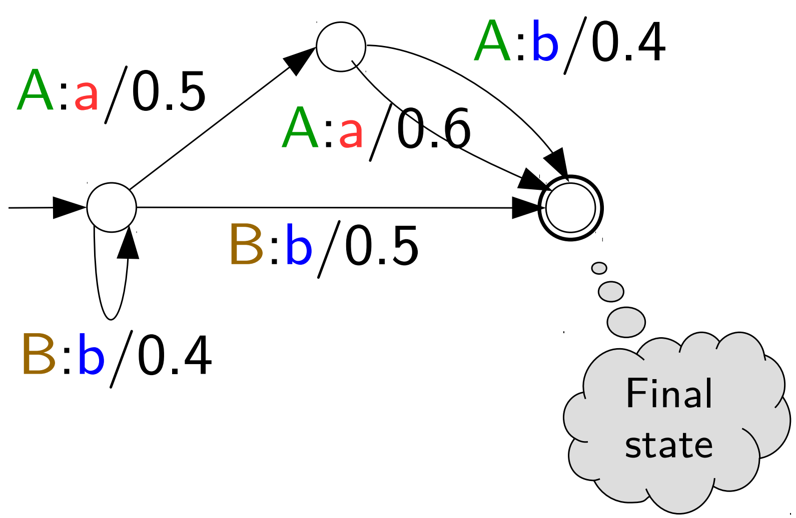
- Until end of word
  - Aggregate acoustics
- At end of word
  - Add LM score and insertion penalty
- At each step
  - Check the number of active states

    → Pruning (a, b, c)



WFST-based decoding

- Weighted Finite State Transducer (WFST)
  - Finite state automaton
  - Weighted edged
  - Output symbol (in addition to input symbol)
- String conversion



Input sequence: BBB

$\rightarrow$ Ouput sequence: bbb

Probability: $0.4 \times 0.4 \times 0.5 = 0.08$

Input sequence: AA

$\rightarrow$ Best ouput sequence: aa

Probability: $0.5 \times 0.6 = 0.3$

$\rightarrow$ 2^{nd} best output sequence: ab

Probability: $0.5 \times 0.4 = 0.2$

WFST-based decoding

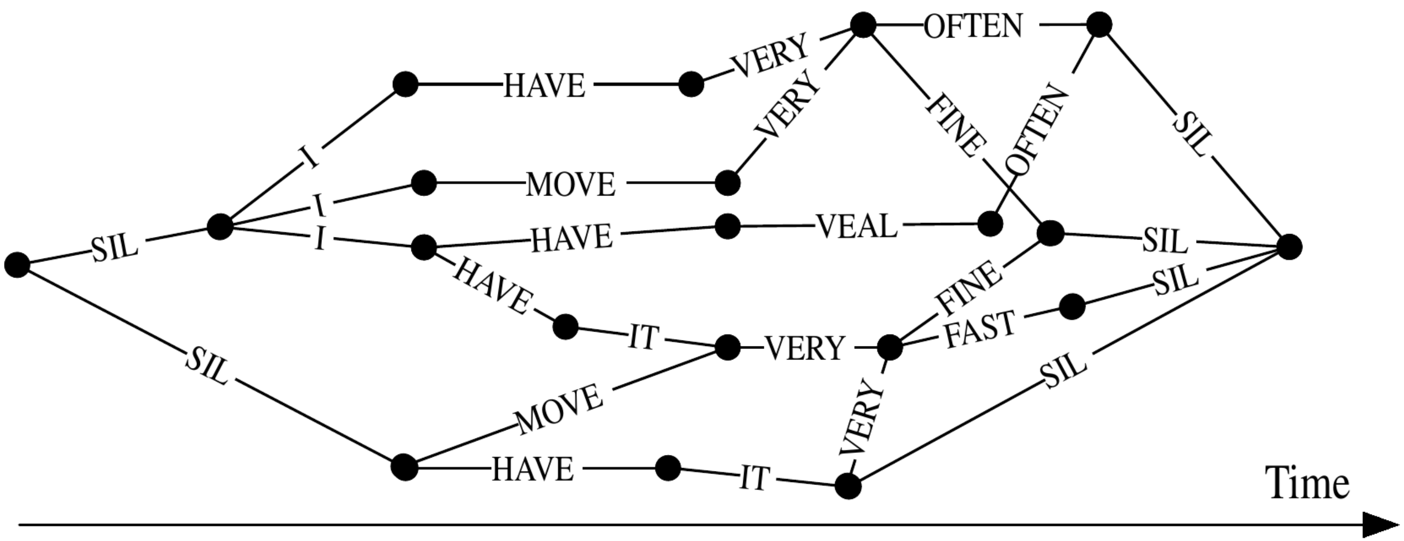
- All models can be written as WFSTs

	transducer	input sequence	output sequence
G	word-level grammar	words	words
L	pronunciation lexicon	phones	words
C	context-dependency	CD phones	phones
H	HMM	HMM states	CD phones

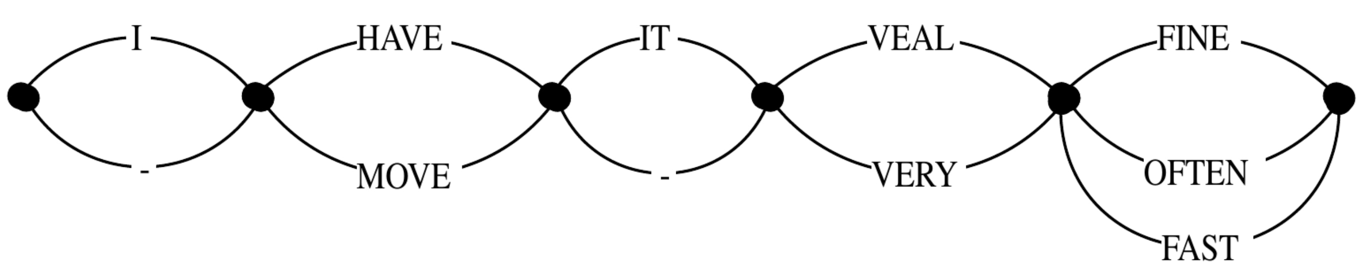
- WFST composition
  - $H \circ C \circ L \circ G$ (+ determinization + minimization) maps HMM states to words
- Fast but requires memory
- Kaldi toolkit

Alternative hypotheses

- Word lattice



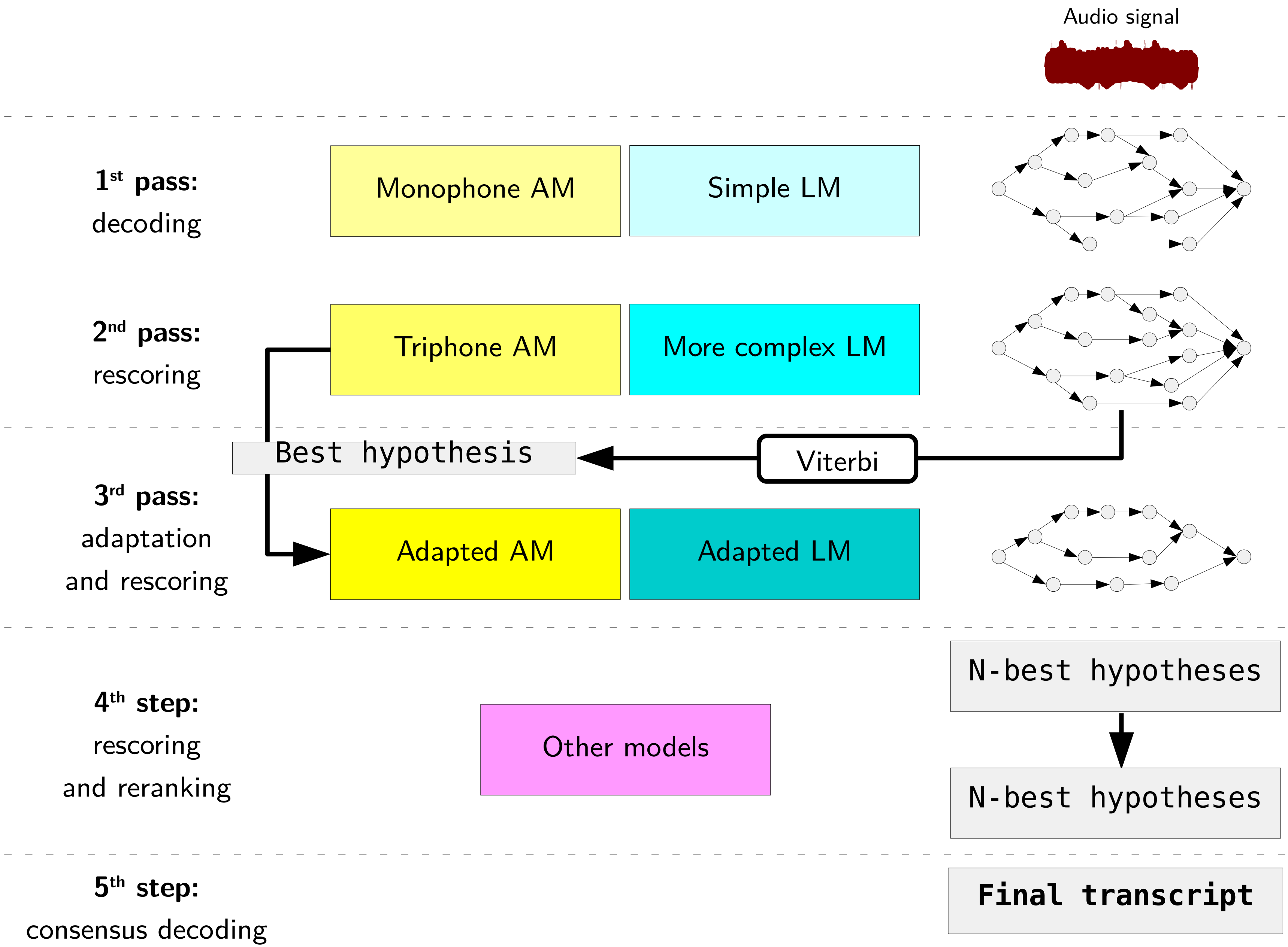
- Confusion network



- Confidence measures
- Rescoring

(Source: Gales and Young, 2008)

Multi-pass architecture



End-to-end approach

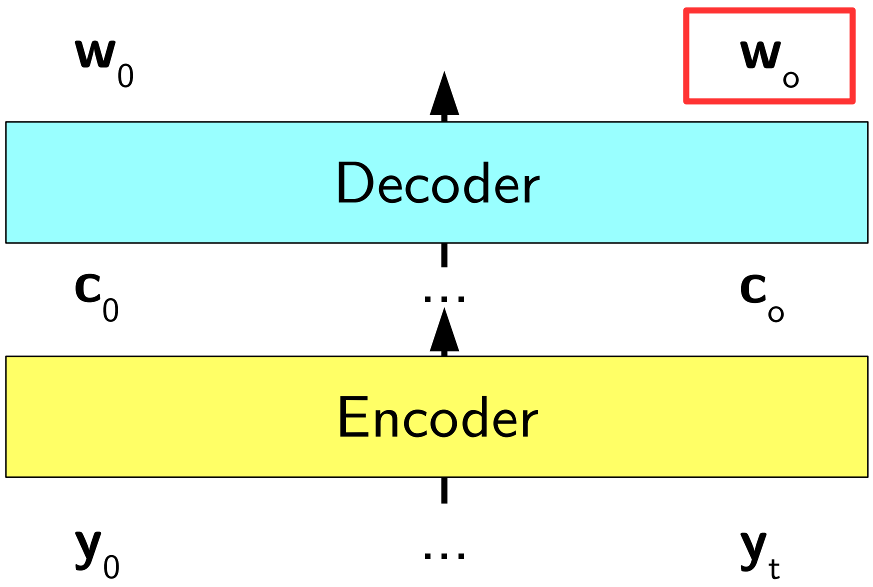
Reading:

- Lu et al. (2015). A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition. In Proc. Interspeech (pp. 3249-3253).

End-to-end approach

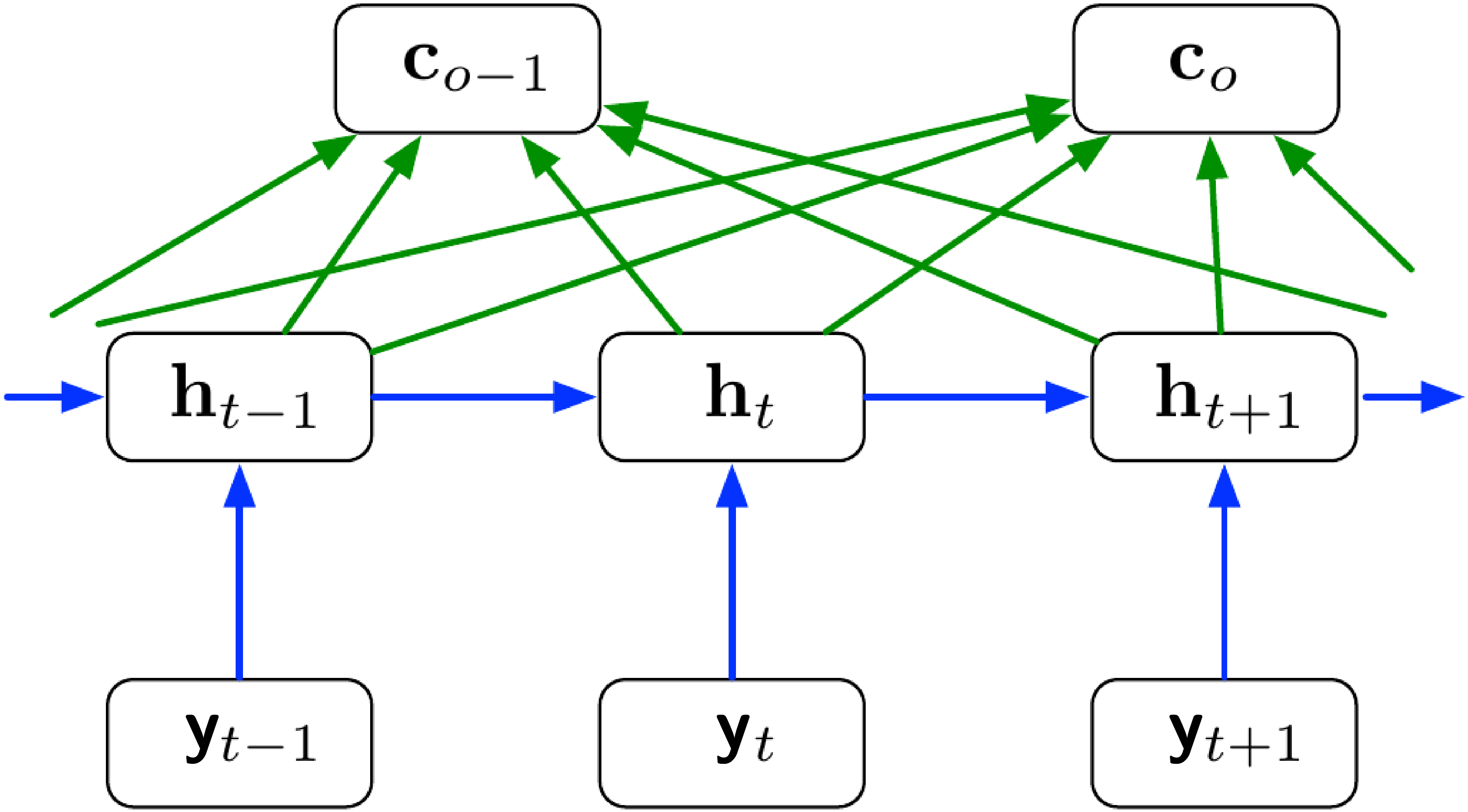
$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \mathsf{P}(\mathbf{W}|\mathbf{Y})$$

- Encode **Y** ($\mathbf{y}_t$) as a sequence of contexts **C** ($\mathbf{c}_t$)
- Decode **C** into a sequence of words **W**



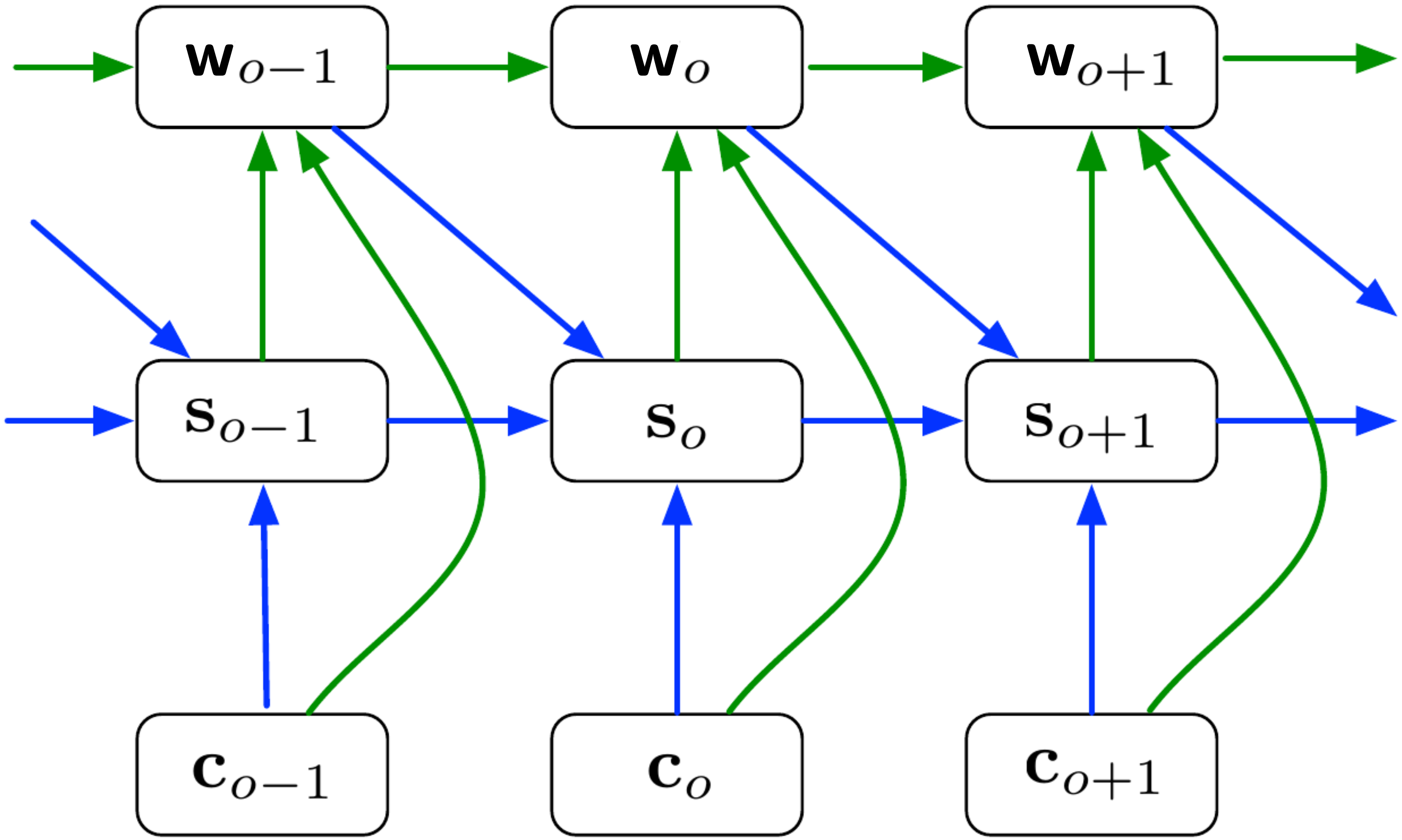
- Beam-search decoding

RNN encoder (source : Renals, 2017)



- $\mathbf{y}_t$, input at step t
- $\mathbf{h}_t$, (encoder's) hidden layer at step t
- $\mathbf{c}_o$, context at step o

RNN decoder (source : Renals, 2017)



- $\mathbf{c_o}$, context at step o
- $\mathbf{s_o}$, (decoder's) hidden layer at step o
- $\mathbf{w_o}$, output at step o

Related tasks

Adaptation

- Speaker adaptation
  - GMM, DNN parameter changes
- Language model
  - (A) Grab task-related texts (web)
  - (B) Spot discriminating words, phrases
  - Increase probs
- Vocabulary
  - (A + B)
  - Sub-word units, phonetic transcription
  - Enrich pronunciation dictionnary
  - Add n-grams / exploit related words parameters

Usage of speech transcripts

- Spoken language understanding/dialogue
- Spying
- Command
- Indexation/information retrieval
- Clustering/summarization
- Multimedia hyperlinking

Conclusion

Keypoints

- Statistical approach
  - Acoustic model, language model
- End-to-end approach
  - 1 big neural network
- Current trends
  - LSTMs
  - Removal of expert (acoustic, linguistic) knowledge
- Remaining challenges
  - Adaptation
  - Noisy environments
- Performance
  - Human still not beaten
  - Many possible applications yet