

# Vorlesung

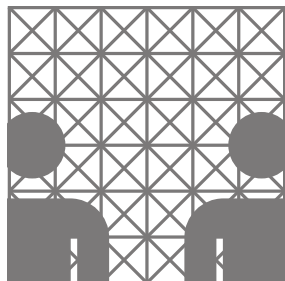
# Sprachdialogsysteme

Timo Baumann  
[baumann@informatik.uni-hamburg.de](mailto:baumann@informatik.uni-hamburg.de)



<https://nats-www.informatik.uni-hamburg.de/SDS20>

Universität Hamburg, Department of Informatics  
Language Technology Group



# Heute

## Spracherkennung in a nutshell

- wie kann ich das Spracherkennungsproblem modellieren, sodass
  - eine Lernbarkeit der Modellparameter anhand von Beispieldaten möglich wird
  - Hidden-Markov-Modell
- wie funktioniert gesprochene Sprache?

# Unser Modell der Kommunikation

Sprecher



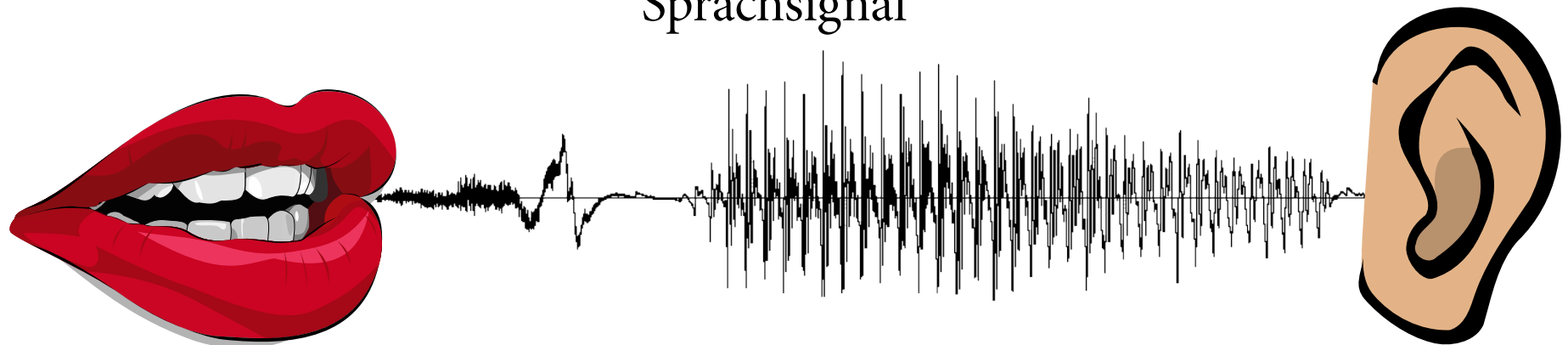
dekodierte  
linguistische  
Representation

sensorischer  
Eindruck

Hörer

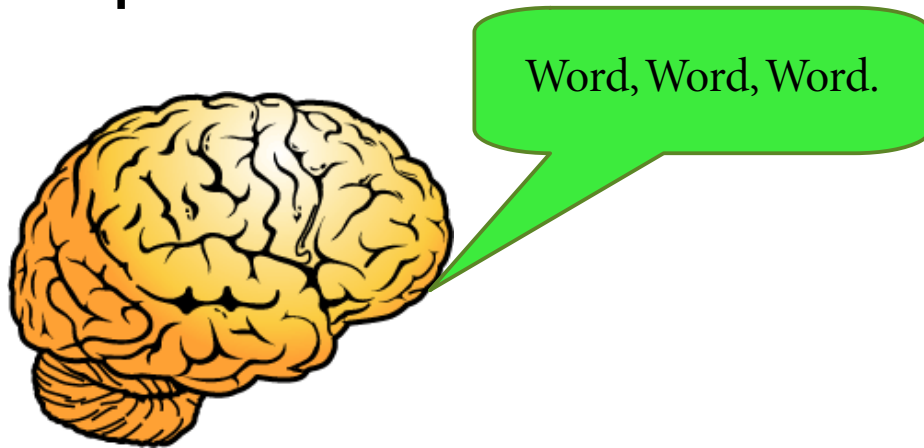


Sprachsignal



# Noisy-Channel Model

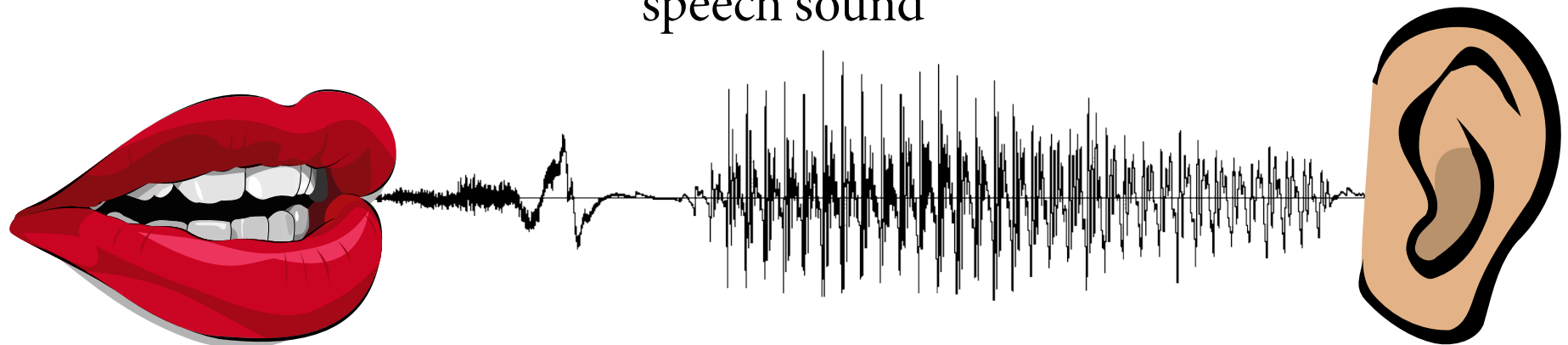
Sprecher



Hörer

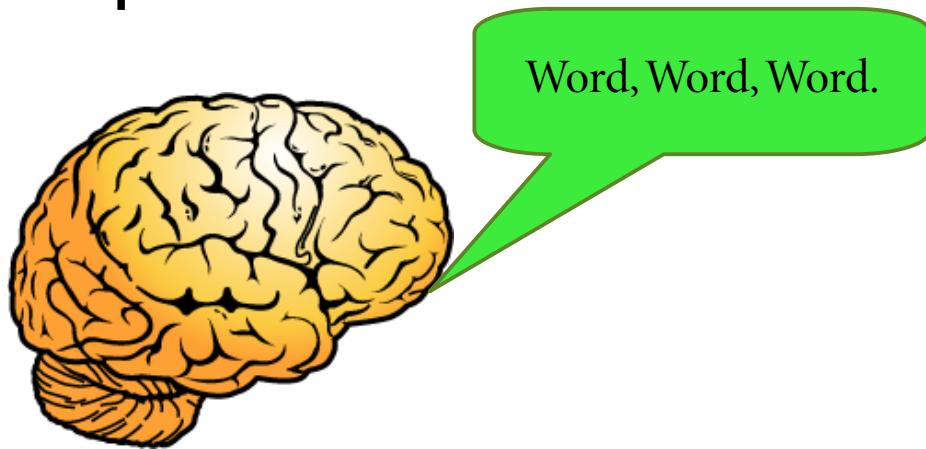


speech sound



# Noisy-Channel Model

Sprecher

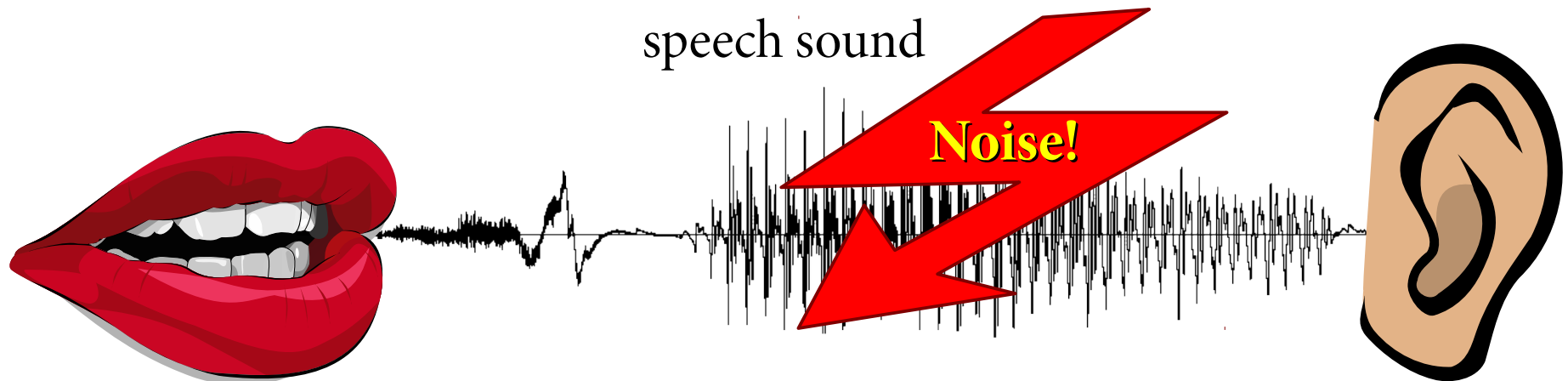


Hörer

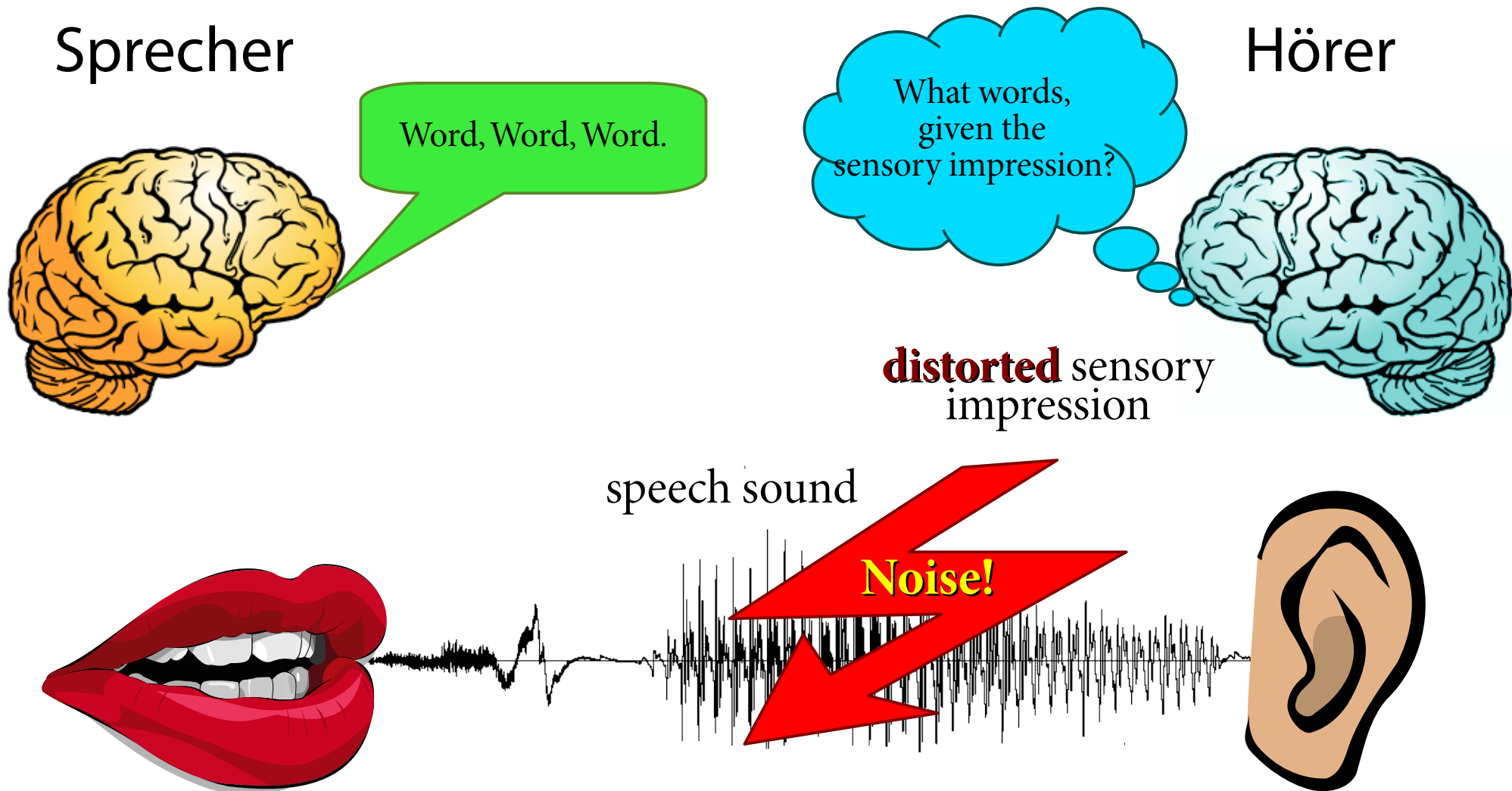


**distorted** sensory  
impression

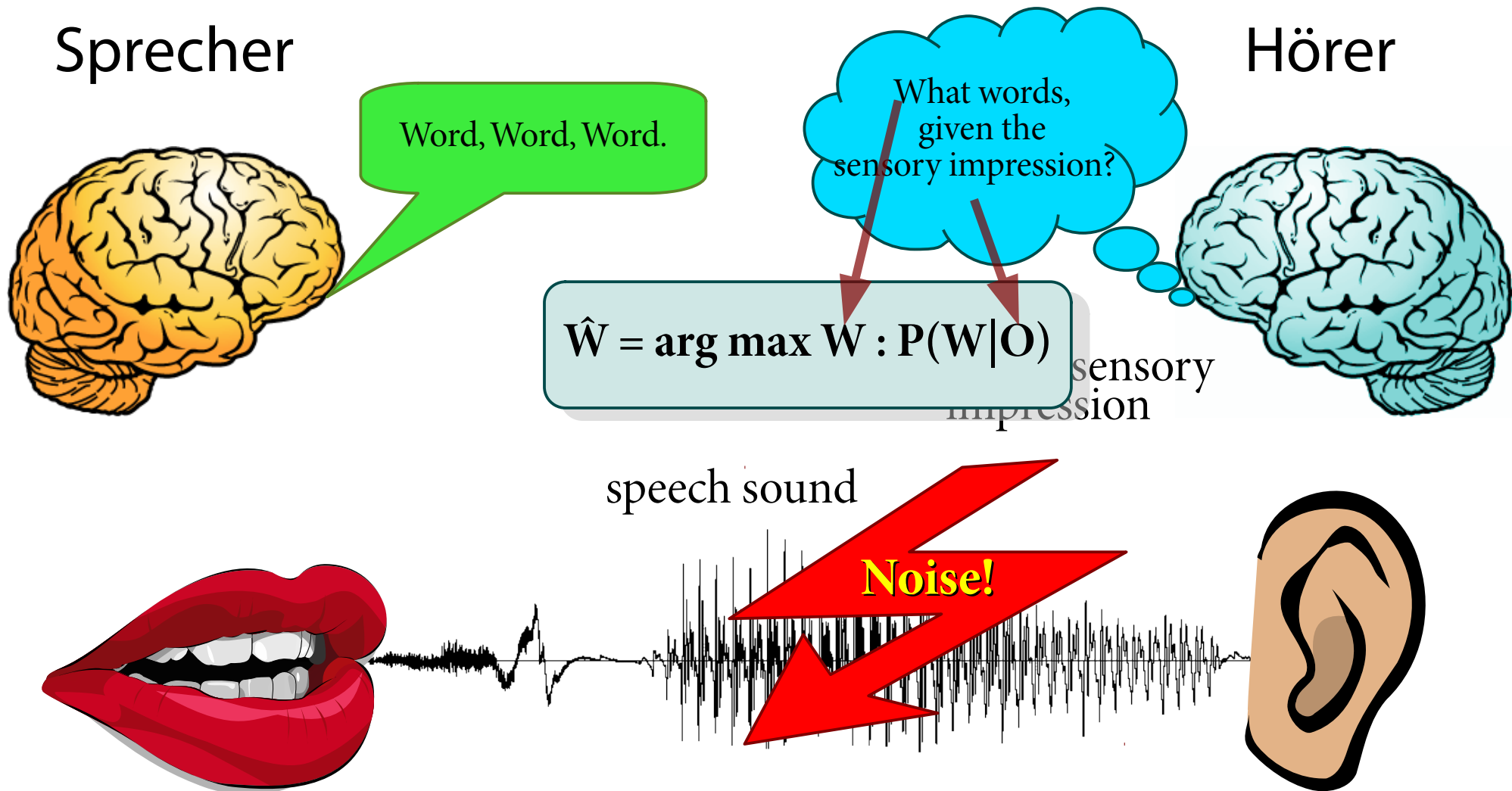
speech sound



# Noisy-Channel Model



# Noisy-Channel Model



# Die Spracherkennungsaufgabe



# Die Spracherkennungsaufgabe

Gegeben eine Sprache  $\mathcal{L}$

- und einen sensorischen Eindruck (Observation)  $\mathbf{O}$ 
  - Parametersequenz die das Sprachsignal (alle 10 ms) beschreiben
- suchen wir  $\hat{\mathbf{W}}$  in  $\mathcal{L}$  mit
  - $\hat{\mathbf{W}} = \arg \max \mathbf{W} : P(\mathbf{W}|\mathbf{O})$   
die am *wahrscheinlichsten* zur Observation passende Wortsequenz
  - “maximum-likelihood principle”
- wie bestimmen wir  $P(\mathbf{W}|\mathbf{O})$ ?
- wie organisieren wir die Suche ( $\arg \max$ )? → Selbststudium

# Bayes'sche Regel

Gegeben zwei bedingte Wahrscheinlichkeiten A und B:

- $$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

$$\hat{W} = \arg \max W : P(W|O)$$

- wir nutzen  $\arg \max$  in unserer Formel  $\rightarrow$  der Nenner  $P(B)$  ist für das Ergebnis irrelevant, daher:
- $P(A|B) \sim P(B|A) \times P(A)$

# Die Spracherkennungsaufgabe (II)

–  $\hat{W} = \arg \max W : P(W|O)$

• Bayes'sche Regel:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

–  $\hat{W} = \arg \max W : \mathbf{P(O|W)} \times \mathbf{P(W)}$

–  $P(O|W)$ : **Akustisches Modell**

- Beobachtungswahrscheinlichkeit für eine gegebene Wortfolge
- *What do words sound like?*

–  $P(W)$ : **Language Model** (Wortfolge-Modell)

- a priori Wahrscheinlichkeit einer Wortfolge
- *What word sequences are likely?*

Anhand welcher Arten von Daten können Sie

(a) das akustische Modell  $P(O|W)$  schätzen,

(b) das Wortfolgemodell  $P(W)$  schätzen?

$$P(O|W)$$

Akustisches Modell auf Basis von Wörtern?

Gibt es eine (noch) angemessenere Basis?

noch      Koch      Loch      Woche      Docht      dicht      nicht

# Words or Phonemes?

- acoustics primarily depend on phonemes, not on words
- words have an internal structure (cmp. previously)
  - this was disregarded in early approaches e.g. for single-word recognition. Hence it's almost always ignored in descriptions.
- thus we should rather estimate  $P(O|Ph)$ , instead of  $P(O|W)$
- we need an additional conversion step that relates words to phoneme sequences  $P(Ph|W)$

# Das Aussprachelexikon als Brücke von akustischem zum Wortfolgemodell

$$\hat{W} = \arg \max W : \mathbf{P(O|Ph)} \times \mathbf{P(Ph|W)} \times \mathbf{P(W)}$$

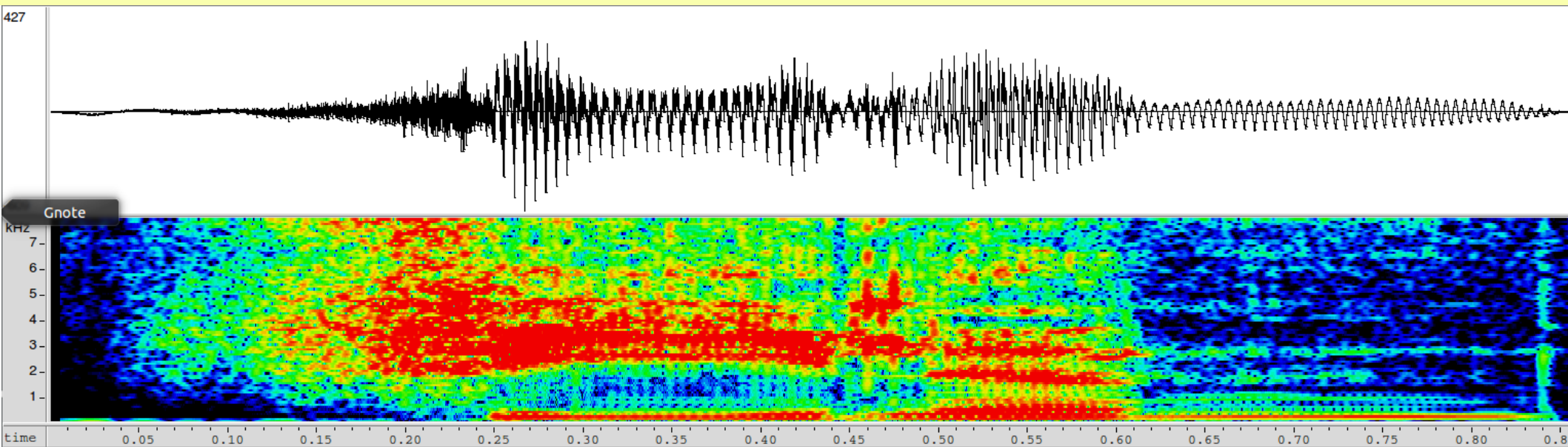
- einfache Wörterbücher mappen jedes Wort auf eine Lautfolge
- Erweiterungen:
  - Aussprachevarianten erfassen
  - Aussprachemodell erlernen (regel-basiert oder sonstwie)
  - Lexikon an Sprecher adaptieren (Tempo, Kontext, Dialect, ...)

# Die Spracherkennungsaufgabe (III)

- $\hat{W} = \arg \max W : \mathbf{P}(\mathbf{O}|\mathbf{Ph}) \times \mathbf{P}(\mathbf{Ph}|W) \times \mathbf{P}(W)$ 
  - die einfachste Form von  $\mathbf{P}(W)$  könnte eine Liste möglicher Sätze und ihrer zugeordneten Wahrscheinlichkeiten sein.
  - $\mathbf{P}(\mathbf{Ph}|W)$  im einfachsten Fall ein Wörterbuch
- das **akustische Modell  $\mathbf{P}(\mathbf{O}|\mathbf{Ph})$** 
  - bewertet das Sprachsignal auf Passfähigkeit zu einer Lautfolge
  - Beschreibung des Signals durch Sequenz von Beobachtungsmerkmalen:
- $\mathbf{O} = (o_1, o_2, o_3, o_4, \dots, o_{t_{\max}})$ ,  
 $o_i$  sind Merkmalsvektoren, die über kurze Audiofenster berechnet werden können (Details: nächste Woche)



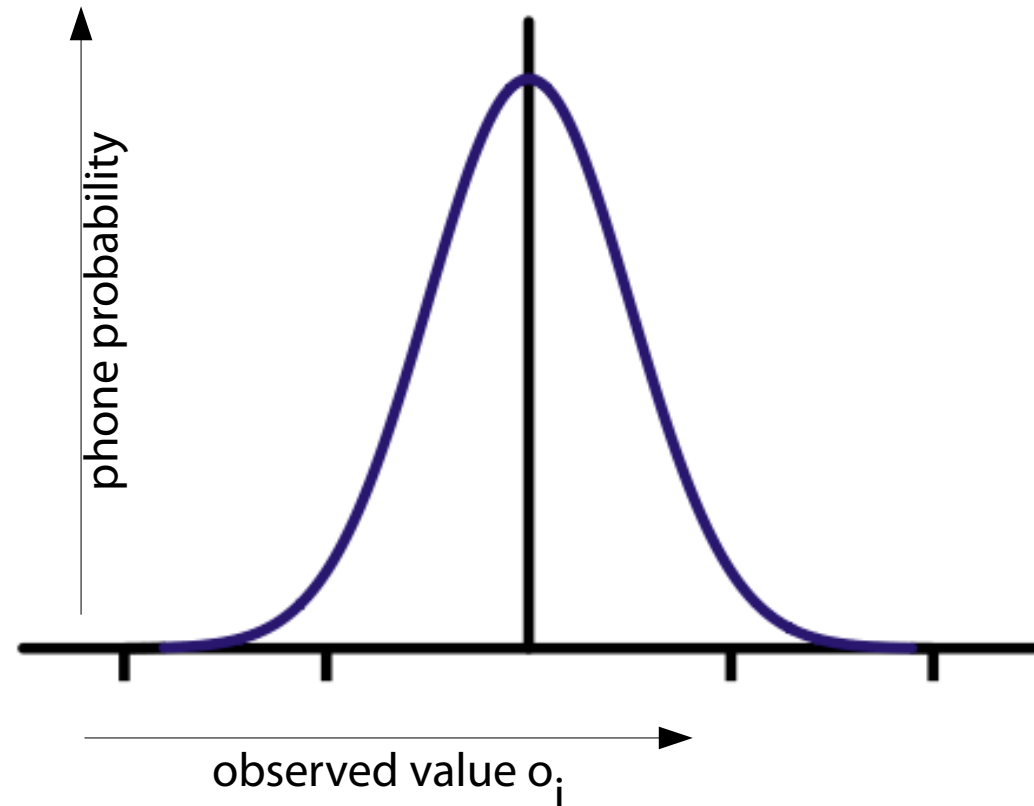
# Spectrogram



- Sprache ist ein “quasi-stationäres” Signal:
  - schnelle sich wiederholende Signaländerungen (Perioden mit einer Grundfrequenz)
  - langsame Änderung der Eigenschaften innerhalb der Perioden
- regelmäßige Betrachtung des Signals (fenstern) über mehrere Perioden hinweg

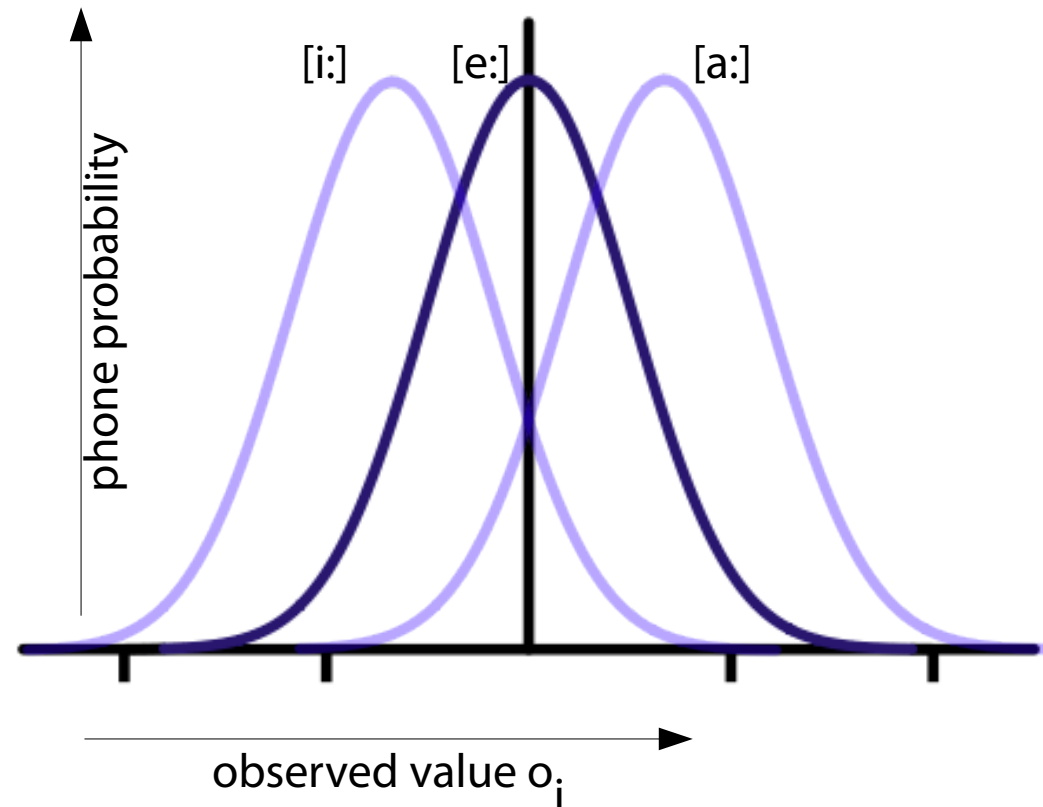
# Von Observierungen zu Wahrscheinlichkeiten

- jeder Laut erhält sein eigenes Modell
- jedem Lautmodell wird eine Akzeptanzfunktion zugeordnet, die der Observation  $o_i$  eine Wahrscheinlichkeit zuordnet
- üblich: Gaußverteilungen
  - nur zwei Parameter:  $\mu$  and  $\sigma$
- Wahrscheinlichkeit lässt sich anhand der Observation schätzen
- $o_i$  könnte zu jedem Laut gehören  
→ Wahrscheinlichkeit für alle Laute berechnen



# Von Observierungen zu Wahrscheinlichkeiten

- jeder Laut erhält sein eigenes Modell
- jedem Lautmodell wird eine Akzeptanzfunktion zugeordnet, die der Observation  $o_i$  eine Wahrscheinlichkeit zuordnet
- üblich: Gaußverteilungen
  - nur zwei Parameter:  $\mu$  and  $\sigma$
- Wahrscheinlichkeit lässt sich anhand der Observation schätzen
- $o_i$  könnte zu jedem Laut gehören  
→ Wahrscheinlichkeit für alle Laute berechnen



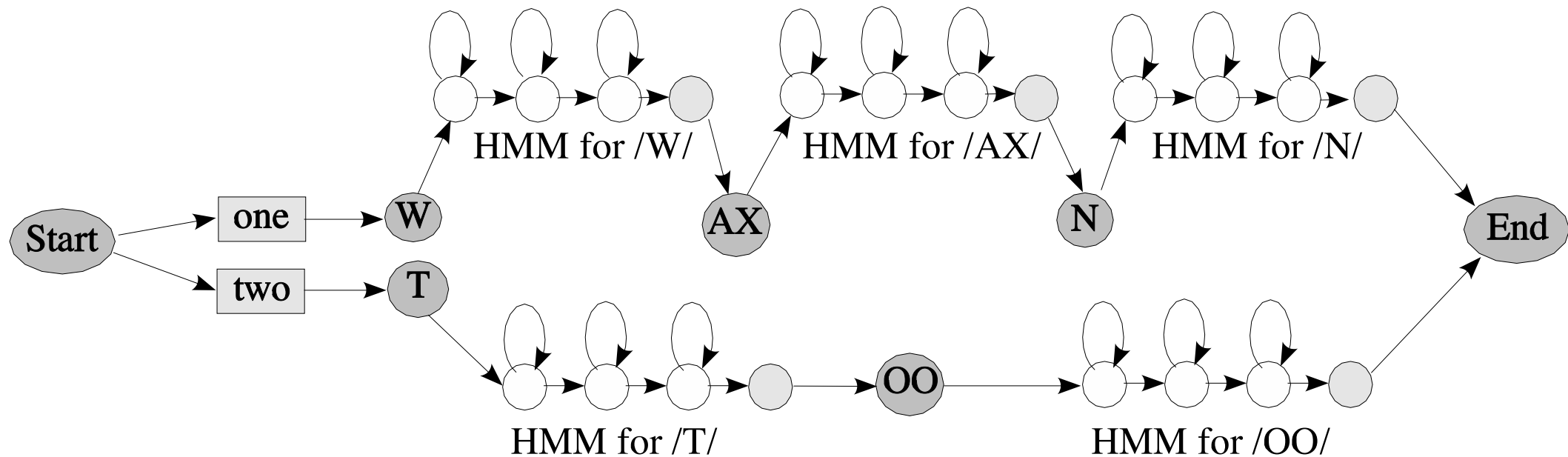
# Lautmodelle

- jedes Lautmodell “akzeptiert” eine Observation mit einer gewissen Wahrscheinlichkeit  
(=wie gut passt das gehörte zu dem Laut)
- meist dauert ein Laut länger als eine Observation
  - aber wie lang genau?
- Modellierung durch Übergangswahrscheinlichkeiten
  - Laute unterscheiden sich in ihrer “Lieblingsdauer”
- transition probabilities + observation probabilities
  - ... plus Lexicon plus Language Model ...
    - Hidden Markov Models to the rescue!

# Hidden-Markov Models

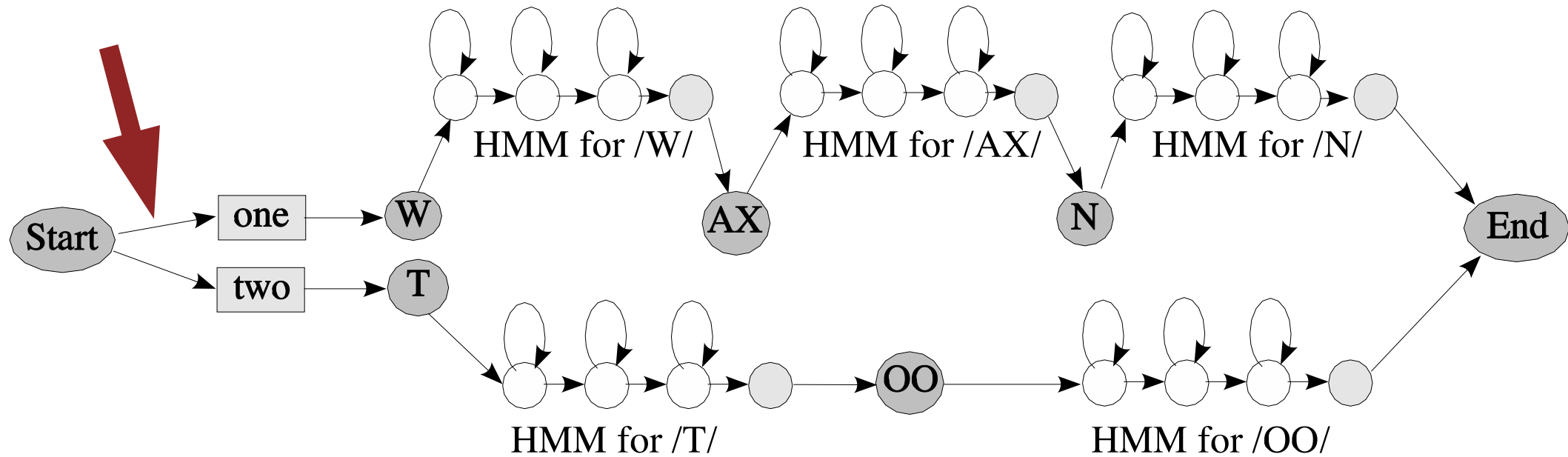
- einheitliches Modell für Spracherkennungsvorgang
- **Markov**-Annahme: die Zukunft hängt nur von einer kurzen Vergangenheit ab
  - bzw.: Vergangenheit kann in einen Zustand gepresst werden
  - Observation kann ohne Betrachtung der vollen Historie “verstanden” werden
- wir konstruieren einen Zustandsgraphen in dem jeder Zustand die gesamte (relevante) Historie zusammenfasst

# The Search Graph



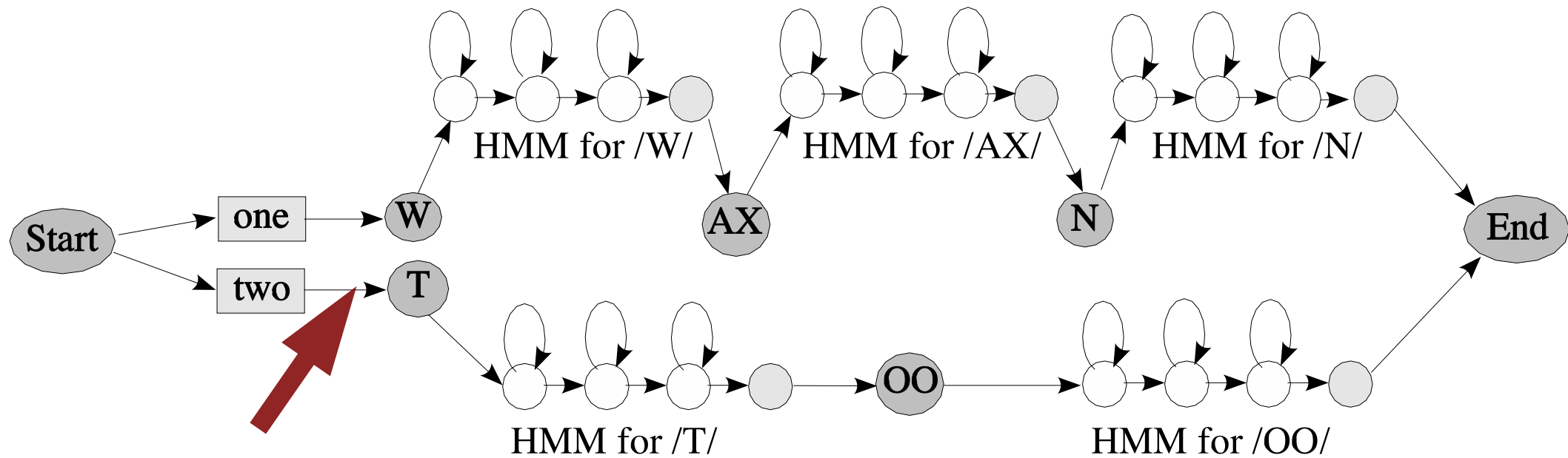
built from language model (here:  $S \rightarrow \text{"one"} \mid \text{"two"}$ ),  
lexicon ( $\text{one} \rightarrow /W AX N/$ ,  $\text{two} \rightarrow /T OO/$ ), and phone models

# The Search Graph



- transition probabilities from language model

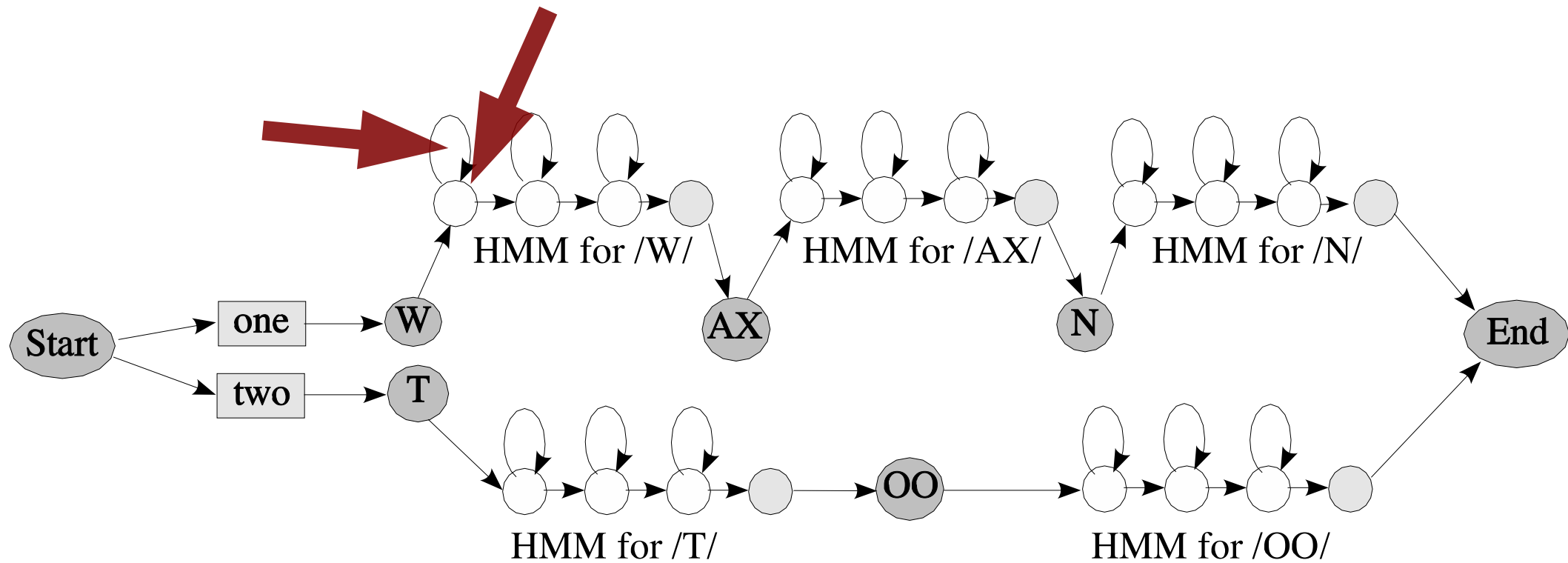
# The Search Graph



- expansion to sounds from the lexicon



# The Search Graph

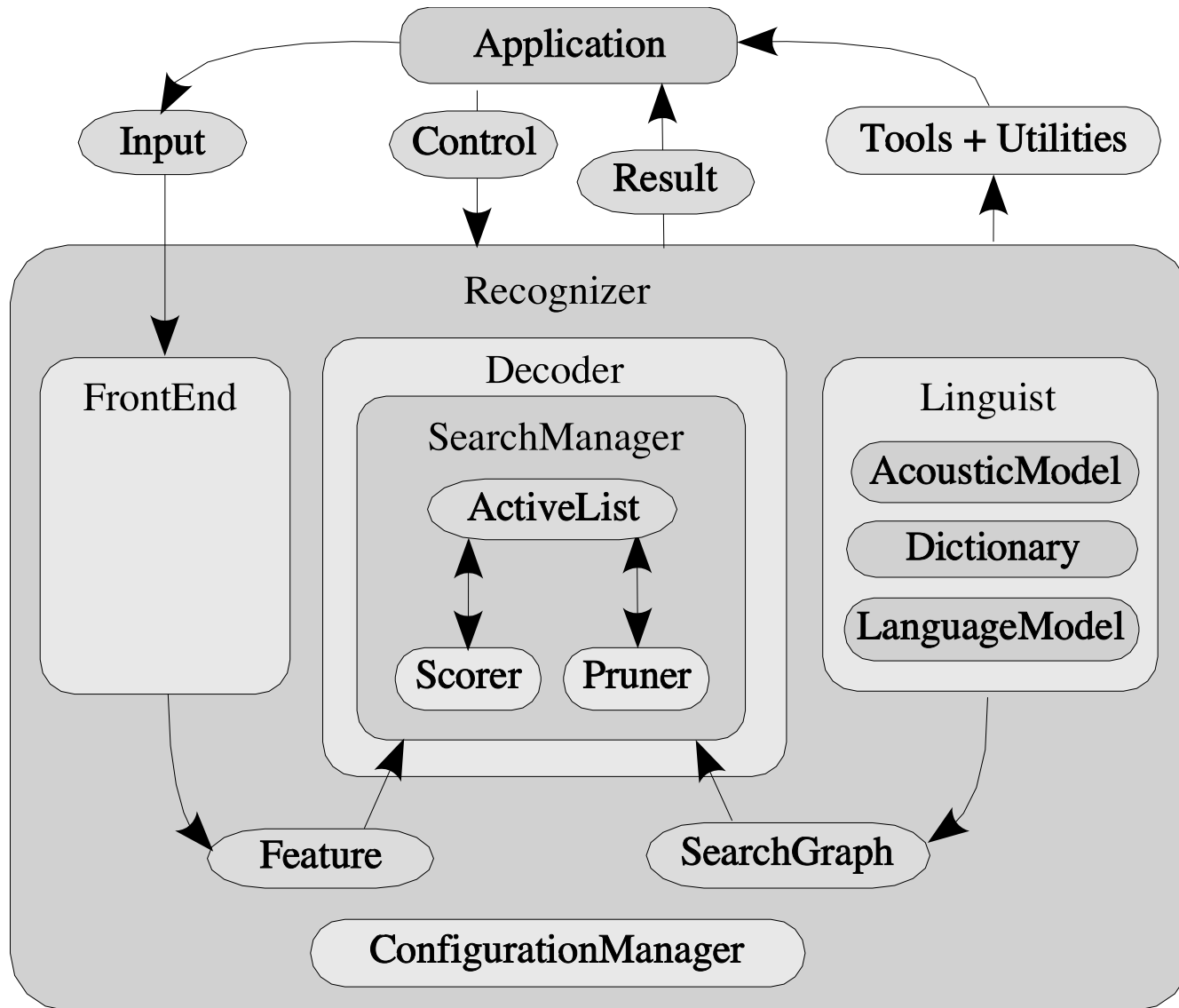


- acoustic model: transition probabilities (A) and emission/observation probabilities (B)

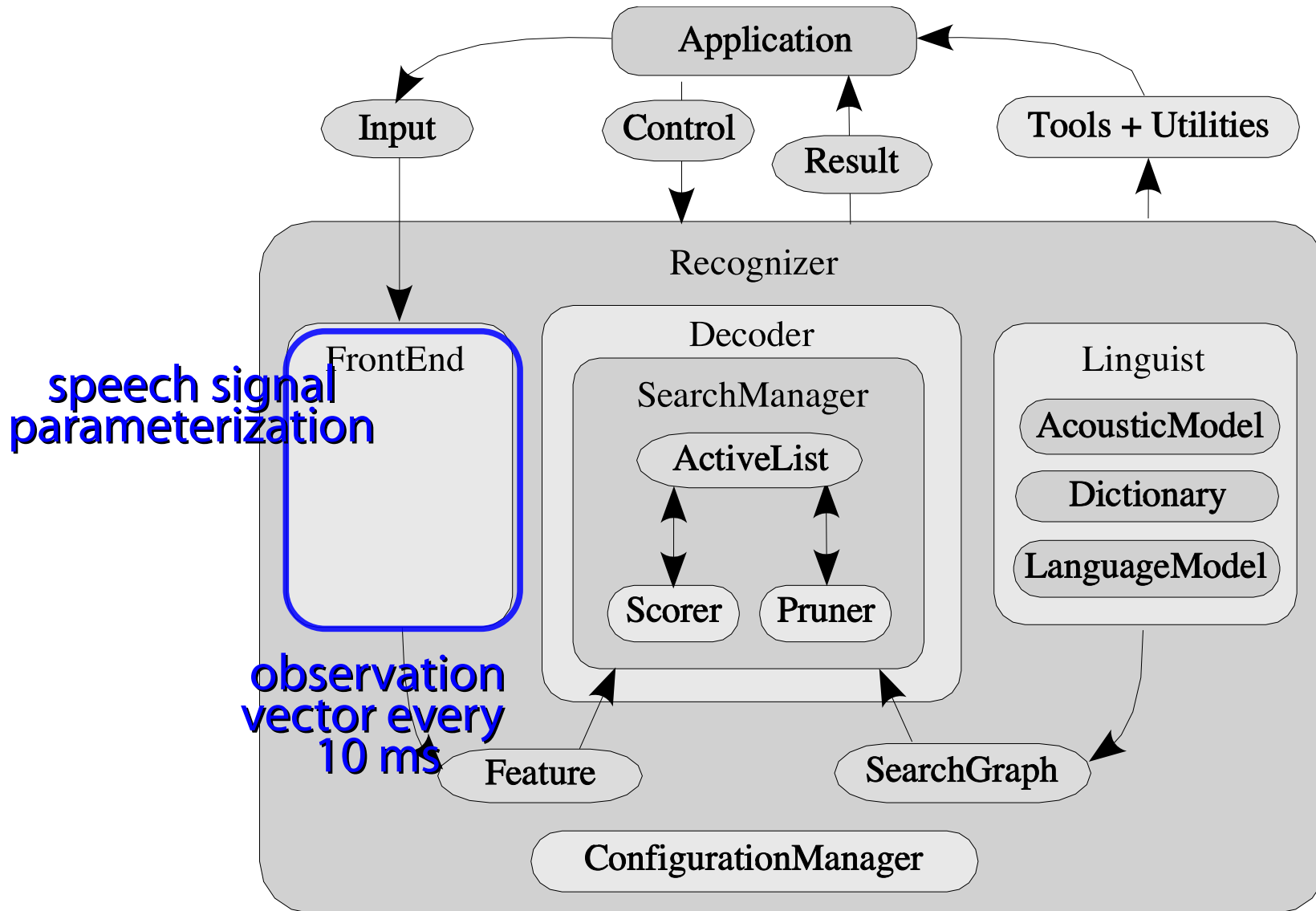
all we need to do is find the most likely  
path through the graph

→ vgl. Leseaufgabe

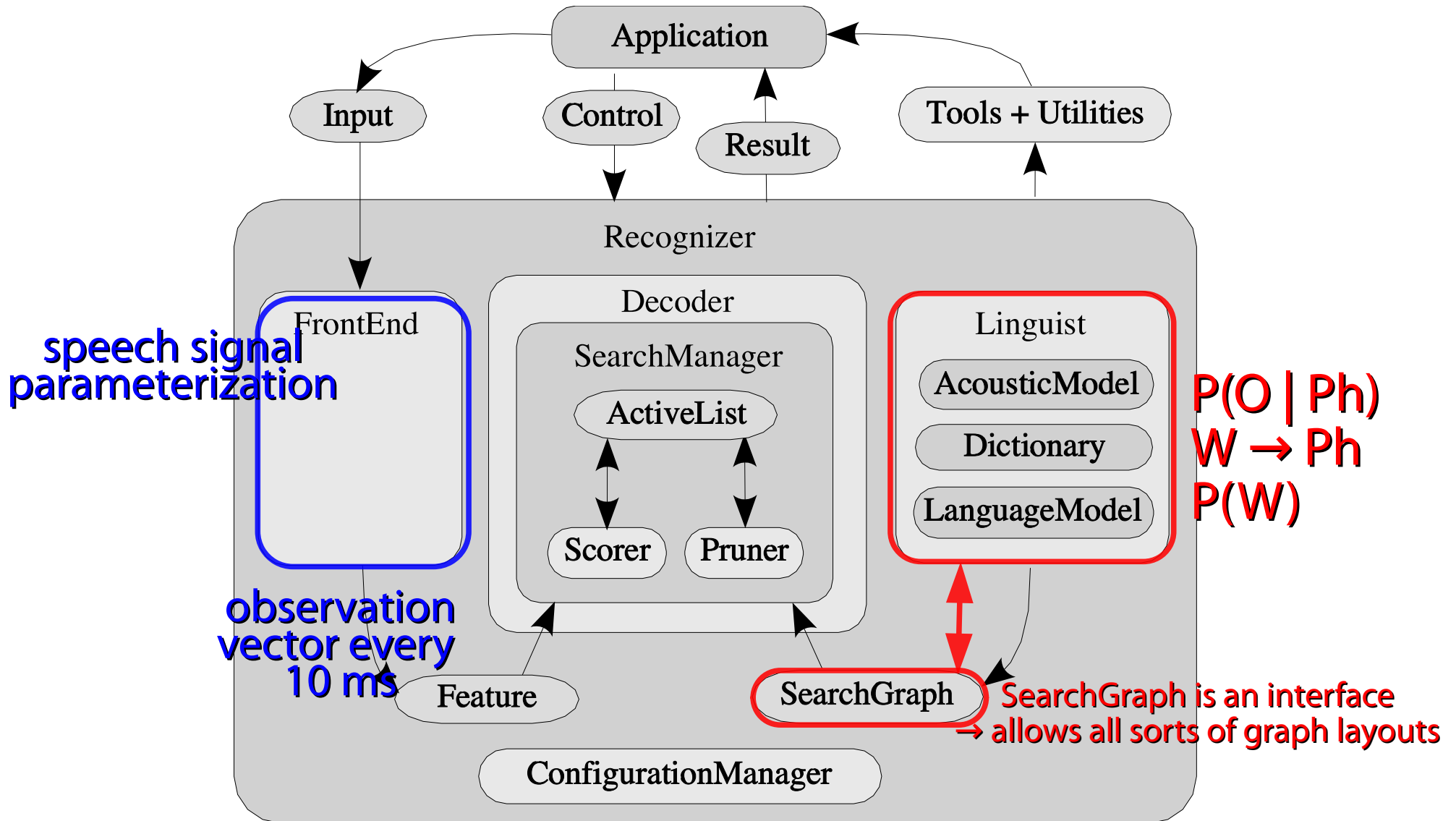
# Sphinx-4: A Flexible Open Source Framework for Speech Recognition



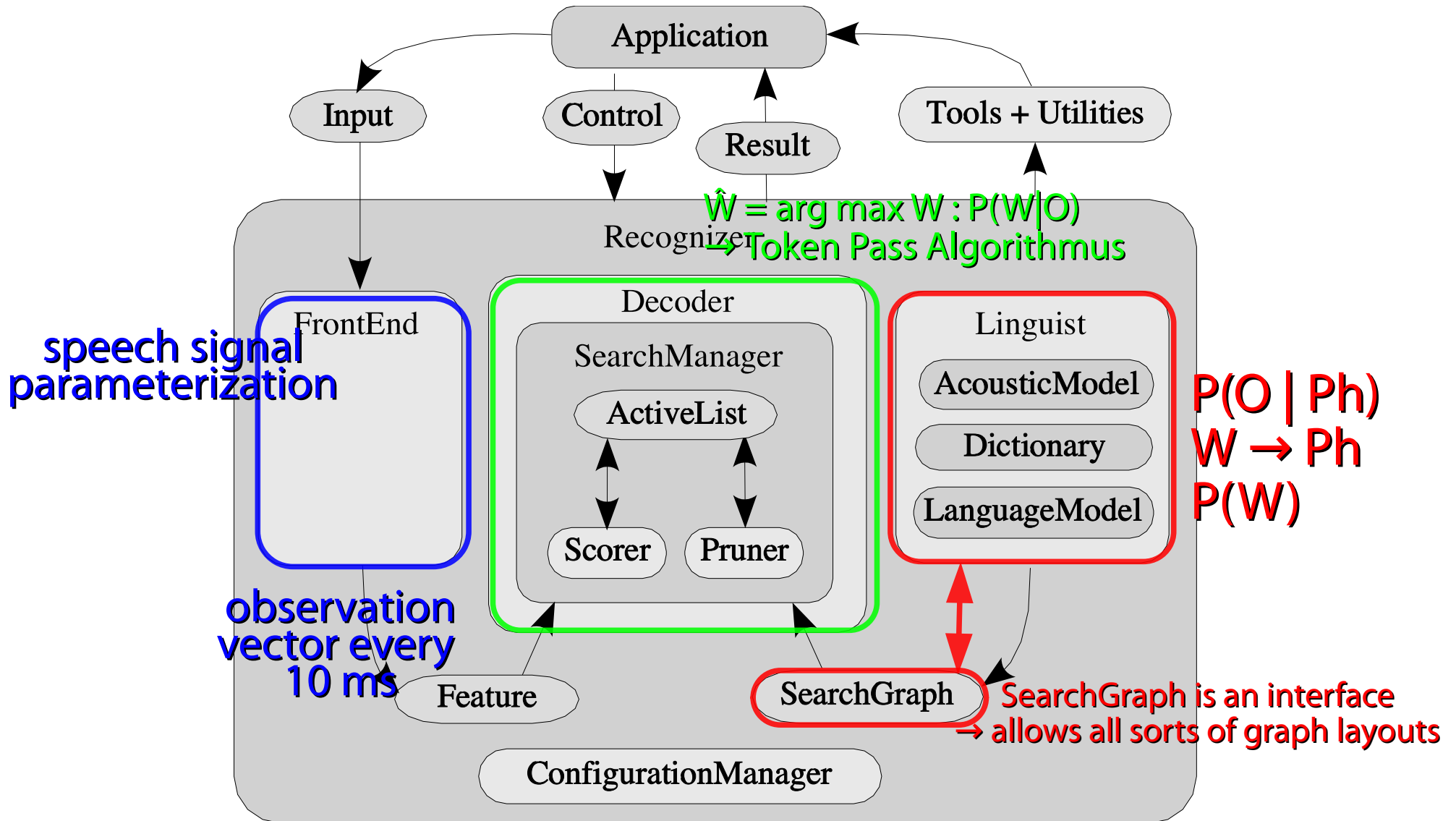
# Sphinx-4: A Flexible Open Source Framework for Speech Recognition



# Sphinx-4: A Flexible Open Source Framework for Speech Recognition



# Sphinx-4: A Flexible Open Source Framework for Speech Recognition



# Zusammenfassung

- Noisy-channel model
- Problem:  $\hat{W} = \arg \max W : P(W|O)$
- Lösung:  $\hat{W} = \arg \max W : P(O|Ph) \times P(Ph|W) \times P(W)$ 
  - $P(W)$ : Wortfolge-Modell  $\rightarrow$  N-Gram, (weighted) Grammar, ...
  - $P(Ph|W)$ : Aussprachemodell  $\rightarrow$  e.g. table lookup, rules, ...
  - $P(O|Ph)$ : akustisches Modell  $\rightarrow$  Hidden Markov Models
- Such-Problem
  - zeitsynchrone Suche, dynamische Programmierung
  - Token-Pass-Algorithmus
- Training vereinfacht durch Auftrennung in:
  - Wortfolgen (kann mit beliebigen Texten trainiert werden)
  - Aussprachelexikon (kann man notfalls von Hand anlegen)
  - Lautmodelle (Audiodateien mit bekanntem Inhalt)

$$\hat{\mathbf{W}} = \arg \max \mathbf{W} : P(\mathbf{W}|\mathbf{O})$$

vs.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

$$\hat{\mathbf{W}} = \arg \max \mathbf{W} : P(\mathbf{O}|\mathbf{W})P(\mathbf{W})$$



# Confidence estimation

- we don't solve the original question  $\arg \max W: P(W|O)$ 
  - hence, we can't use the probability to say how confident we are
  - we do this because  $P(O)$  is untractable to compute and we need to use Bayes' rule
- come up with a heuristic to generate a *confidence measure/rejection threshold* (per sentence or better per word)
  - based on search parameters, acoustic parameters, language model probabilities, dialogue state, multi-modal information, confusion matrices, ...
  - highly useful for downstream processing: „Sorry, I am unsure: did you say Dallas Airport or Dulles Airport in DC area?“ more useful than „Sorry, I am unsure, can you repeat please?“ which is more useful than „Ok, I'll look for flights to Dallas.“

Training and decoding optimizes for  $P(\mathbf{W}|\mathbf{O})$ .

What does this mean?

What could/should be done differently?

- not all utterances are equally important
- we do not typically care for how many utterances we get right, but for the proportion of words that we get right
- but not even all words are equally important
- we have large corpora for speech+text, but little interactional data → hard to optimize for specific types of interaction



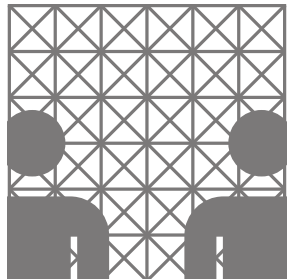
Vielen Dank.

[baumann@informatik.uni-hamburg.de](mailto:baumann@informatik.uni-hamburg.de)



<https://nats-www.informatik.uni-hamburg.de/SDS20>

Universität Hamburg, Department of Informatics  
Language Technology Group



# Notizen

# Further Reading

- the relevant chapters in: Jurafsky and Martin (2009): *Speech and Language Processing*. Pearson International. InfBib: A JUR 4204x.
- Phonetics:
  - J. Neppert (1999): *Elemente einer akustischen Phonetik*. Buske.
- Speech Signal Representation:
  - P. Taylor (2009): *Text-to-Speech Synthesis*. Cambridge Univ Press. ISBN: 978-0521899277. InfBib: A TAY 43070 (accessible introduction to the topic)
  - Rabiner & Juang (1993): *Fundamentals of Speech Recognition*. Prentice Hall. Stabi: A 1994/994. (in-depth mathematical approach)
  - Dong Yu, Li Deng (2015): *Automatic Speech Recognition: A Deep Learning Approach*. Springer. InfBib: A AUT 51465 (NN-based methods)
- Token-Pass Algorithm:
  - Young, Russel, Thornton (1989): “Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems”, *Tech.Rep. CUED/F-INFENG/TR*, Cambridge University.
- The Sphinx-4 Speech Recognizer:
  - Walker et al. (2004): “Sphinx-4: A Flexible Open Source Framework for Speech Recognition”, *Tech.Rep. SMLI TR2004-0811*, Sun Microsystems.

# Desired Learning Outcomes

- understand the optimization target of speech recognition and see implications on the whole-system perspective
- understand how separation of problem into sub-problems
  - leads to better learnability of system parameters
  - leads to suboptimal results
- know and understand the details of the basic speech decoding algorithm based on token-passing, as well as be able to discuss its properties (based on homework)
- understand implications of ASR performance on the whole-system perspective