Projektgruppe


Enes Yigitbas

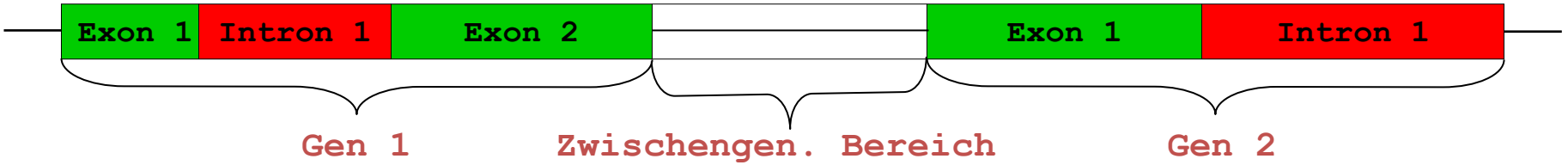
# Text Labeling mit Sequenzmodellen

4. Juni 2010

# Motivation

- Möglichkeit der effizienten Verarbeitung von riesigen Datenmengen
- In vielen Bereichen erwünschte automatisierte Aufgabe: Kennzeichnen bzw. Labeln einer beobachteten Sequenz
- **Anwendungbeispiele:**

(1) Bioinformatik: Labeln von Gensequenzen




Klassifizierung einer gegebenen DNA-Sequenz in kodierenden (Exons), nicht kodierenden (Introns) und Zwischengenetischen Bereich

# Motivation

## (2) Natürliche Sprachverarbeitung: kategorisieren und analysieren von Texten

- Viele Probleme in dieser Kategorie können auf das Text Labeling Problem zurückgeführt werden
- Text Labeling ist eine allgemein nützliche Aufgabe zur Lösung von diversen Problemen im Umgang mit natürlichsprachlicher Text (Part-of-speech Tagging, Named Entity Recognition)
- Beispiel (Part-of-speech Tagging):

  Der Ball rollt. → Der (Artikel) Ball (Nomen) rollt (Verb).

  Wörter in einem Text werden sequentiell mit den zugehörigen Wortarten getaggt

# Inhaltsübersicht

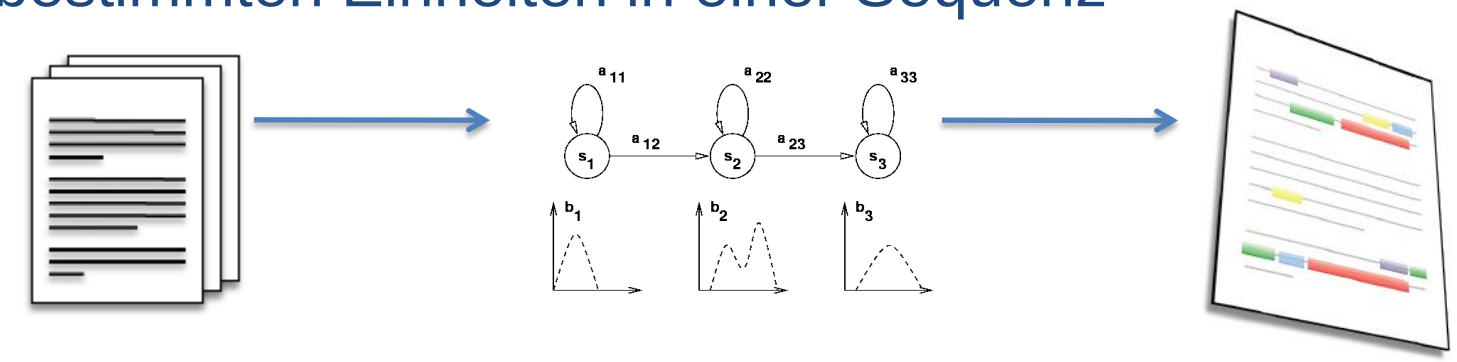
1) Einleitung
2) Grundlagen von Hidden Markov Modellen (HMM)
   - Markovketten
   - Hidden Markov Modell
3) Anwendung von HMMs in der Sprachverarbeitung
   - Decodierungsproblem
   - Viterbi-Algorithmus
4) Zusammenfassung

# Einleitung

Was ist „Text Labeling mit Sequenzmodellen“?

- Einsatz statistischer Prozessmodelle innerhalb maschineller Lernverfahren
- Text Labeling ist eine mögliche Form der sequentiellen Klassifikation

→ Einsatz von Modellen zum Labeln bzw. klassifizieren von bestimmten Einheiten in einer Sequenz



# Einleitung

- Was ist die Aufgabe von statistischen Prozessmodellen?

  Für eine gegebene Sequenz aus Einheiten (Buchstaben, Wörter oder Sätze) Wahrscheinlichkeitsverteilung für mögliche Labels berechnen und die beste mögliche Label Sequenz auswählen.

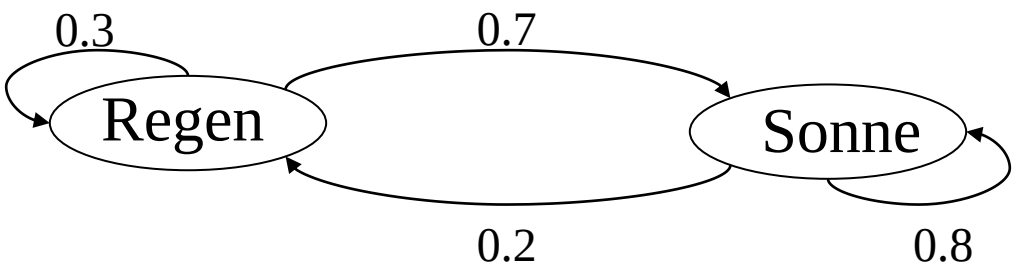
- Welche statistischen Prozessmodelle gibt es?

  - **Hidden Markov Model (HMM)**
  - Maximum Entropy Markov Model (MEMM)
  - Conditional Random Fields (CRF)

  …

# Grundlagen: Markovketten

- Eine ***Markovkette*** ist ein stochastischer Prozess, der nacheinander eine Reihe von Zuständen mit einer gewissen Wahrscheinlichkeit (WK) durchläuft.
- Dabei hängt die WK für den jeweils nächsten Zustand nur vom aktuellen Zustand ab (Markov-Eigenschaft für Markovketten erster Ordnung)
- Markovkette kann als gewichteter endlicher Automat dargestellt werden:

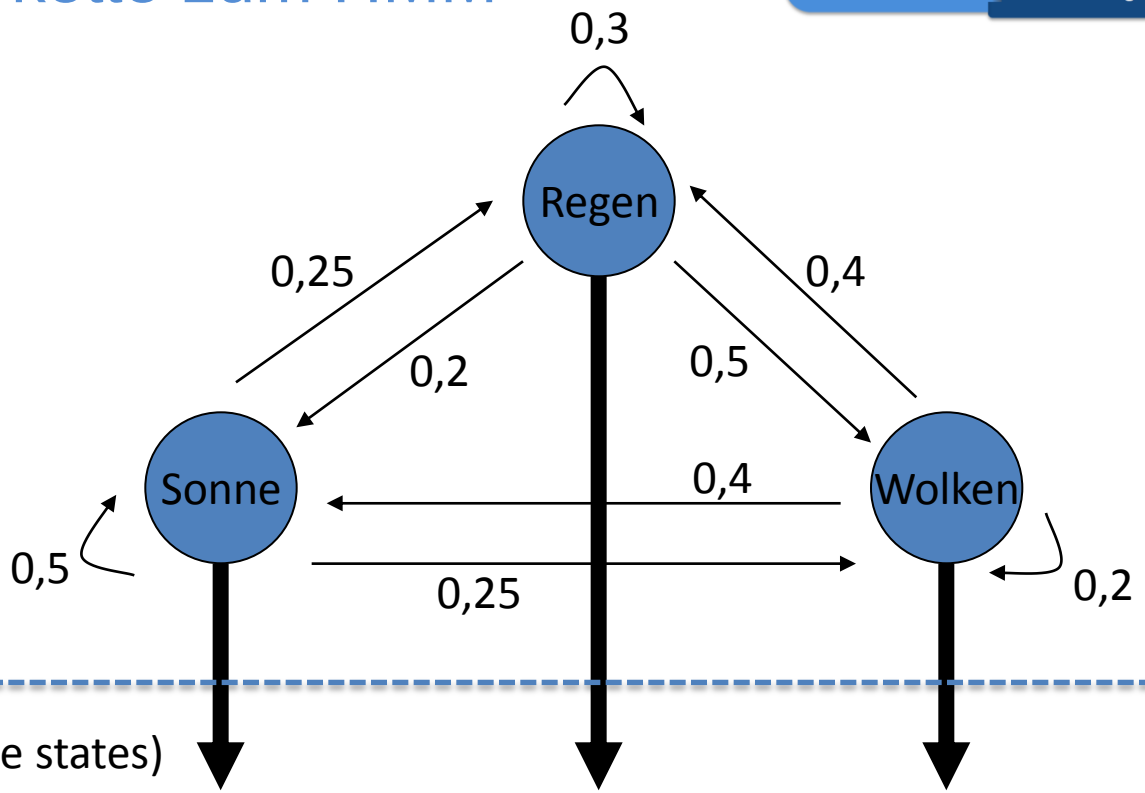


# Hidden Markov Model

- Ein **Hidden Markov Model** ist ein stochastisches Modell, das sich durch zwei Zufallsprozesse beschreiben lässt
- Der erste Zufallsprozess entspricht dabei einer **Markovkette**
- Die Zustände der Markovkette sind von außen jedoch nicht direkt sichtbar (sie sind **verborgen→hidden**)
- Stattdessen erzeugt ein zweiter Zufallsprozess zu jedem Zeitpunkt **beobachtbare** Ausgangssymbole gemäß einer zustandsabhängigen WK-Verteilung

# Von der Markovkette zum HMM

Verborgene Zustände
(hidden states)




Beobachtung (observable states)

| | | |
|---|---|---|
| Trocken: 0,6 | Trocken: 0,05 | Trocken: 0,25 |
| Eher trocken: 0,2 | Eher trocken: 0,1 | Eher trocken: 0,25 |
| Eher feucht: 0,15 | Eher feucht: 0,35 | Eher feucht: 0,25 |
| Feucht: 0,05 | Feucht: 0,5 | Feucht: 0,25 |

# Formale Definition von HMMs:

- HMM als Quintupel $\lambda=(X,A,Y,B,\pi)$

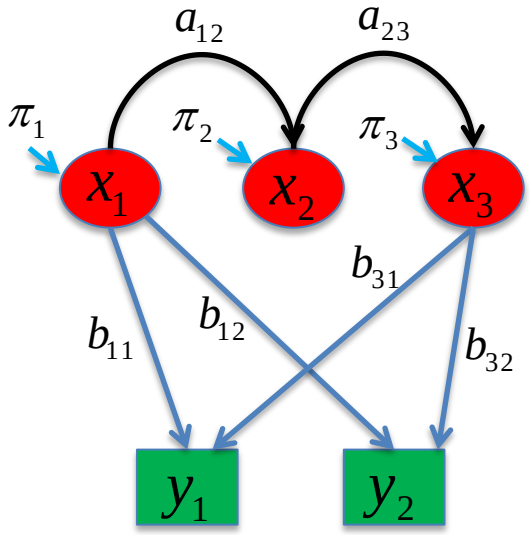
$X = \{x_1, x_2, ..., x_n\}$ Zustände

$A = \{a_{ij}\}$ Zustandsübergangsmatrix

$Y = \{y_1, ..., y_m\}$ Beobachtungen (Emissionen)

$B = \{b_{ij}\}$ Beobachtungsmatrix

$\pi$ Anfangs-WK-Verteilung für Startzustand




(Bsp. Parameter für HMM)

# Beispiel: HMM Matrizendarstellung

- Anfangswahrscheinlichkeit:

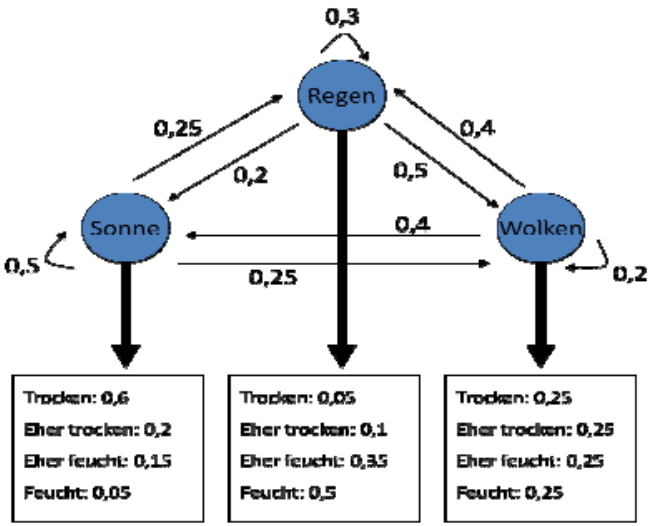
  Vorgegebene Werte, z.B. aus statistischen Daten

  $\pi = (0.63 , 0.17 , 0.2 )$

- Zustandsübergangsmatrix:

$A = \{ a_{ij} \} =$

| | Sonne | Wolken | Regen |
|---|---|---|---|
| Sonne | 0,5 | 0,25 | 0,25 |
| Wolken | 0,4 | 0,2 | 0,4 |
| Regen | 0,2 | 0,5 | 0,3 |




- Beobachtungsmatrix:

$B = \{ b_{ij} \} =$

| | Trocken | Eher Trocken | Eher Feucht | Feucht |
|---|---|---|---|---|
| Sonne | 0,6 | 0,2 | 0,15 | 0,05 |
| Wolken | 0,25 | 0,25 | 0,25 | 0,25 |
| Regen | 0,05 | 0,1 | 0,35 | 0,5 |

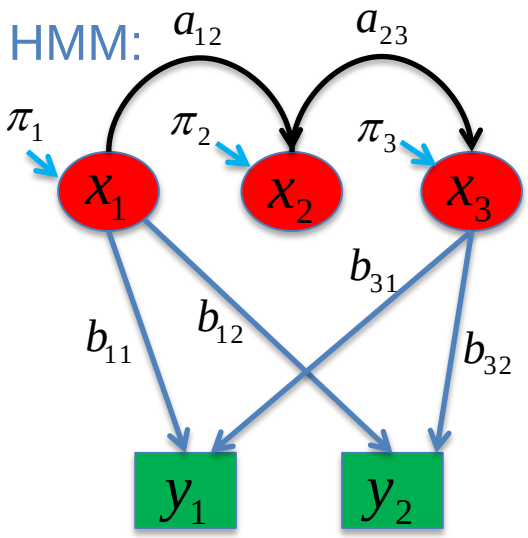
# Grundlegende Aufgaben auf einem HMM:

Evaluation / Likelihood:

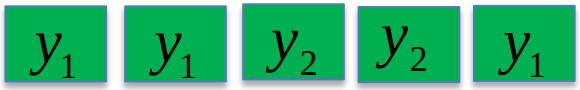
Gegeben: HMM λ, Beobachtungsfolge Y
Gesucht: Wahrscheinlichkeit P(Y| λ)

Beispiel:

Gegeben: (1) HMM:




(2) Beobachtungsfolge:




Frage: Wie groß ist die WK, dass (2) aus (1) generiert wurde?

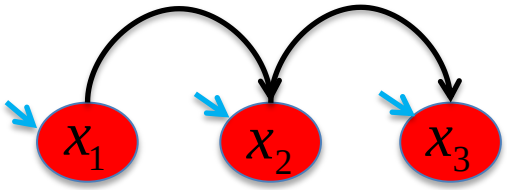
# Grundlegende Aufgaben auf einem HMM:

Learning / Training:

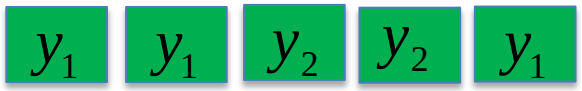
Gegeben: Beobachtungsfolge Y, Zustände X der HMM
Aufgabe: erlerne HMM Parameter A und B, die am wahrscheinlichsten die Beobachtungsfolge Y erzeugen

Beispiel:

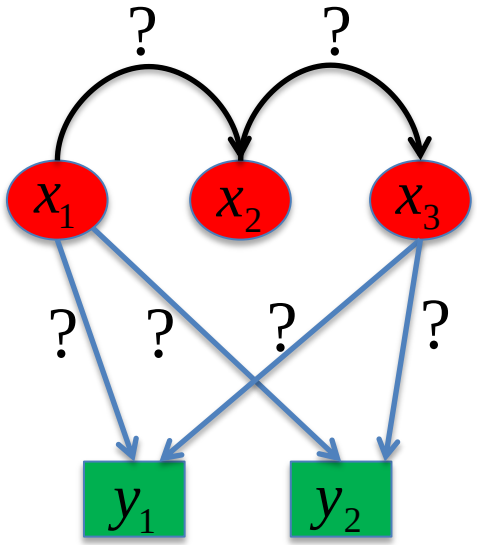
Gegeben:




und




Gesucht:

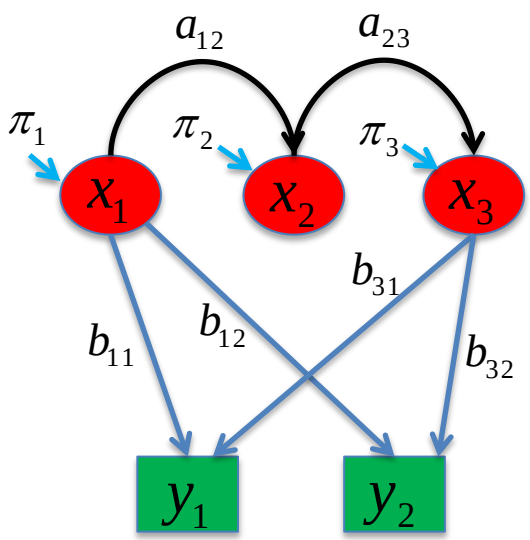


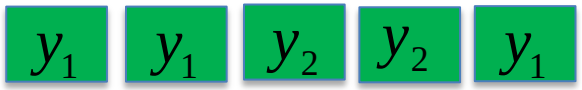
# Grundlegende Aufgaben auf einem HMM:

Decoding:

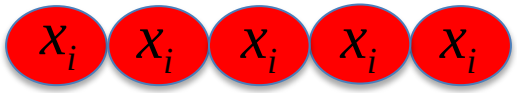
Gegeben: HMM λ, Beobachtungsfolge Y
Gesucht: wahrscheinlichste Sequenz der verborgenen Zustände, die die Beobachtungsfolge Y erzeugt

Beispiel: Gegeben: (1) HMM: (2) Beobachtungsfolge:




Gesucht: wahrscheinlichste Sequenz die (2) erzeugt



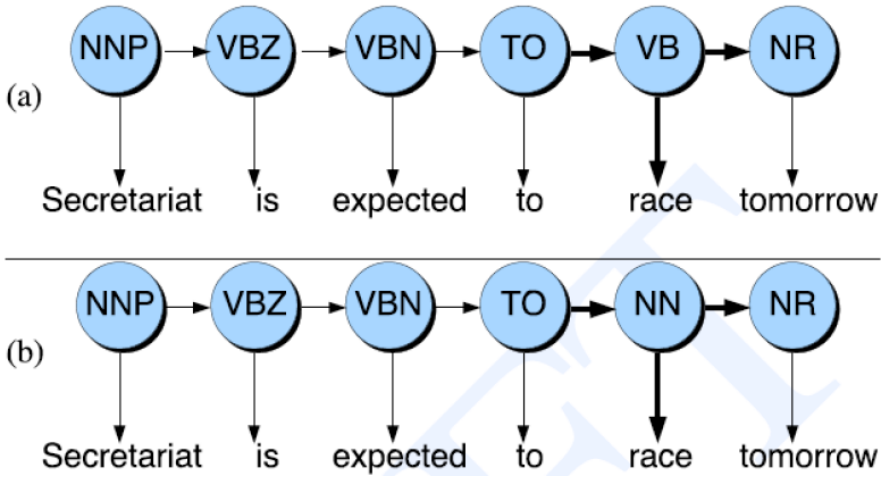

$i \in \{1,2,3\}$

# POS-Tagging und das Dekodierungsproblem

Decoding:

Gegeben: HMM λ, Beobachtungsfolge Y

Gesucht: wahrscheinlichste Sequenz der verborgenen Zustände, die die Beobachtungsfolge Y erzeugt




Welche Sequenz ist wahrscheinlicher ???

# Lösungen für das Dekodierungsproblem

- Wir wollen denjenigen Pfad durch das HMM, der die höchste Gesamtwahrscheinlichkeit hat
- **Naiver Ansatz**: Brute Force
  - Wir probieren einfach jeden möglichen Pfad aus, berechnen die Gesamtwahrscheinlichkeit und wählen am Ende den besten Pfad aus.
  - Problem: extrem ineffizient (exponentielle Laufzeit!)
- **Besserer Ansatz**: Dynamisches Programmieren
  - Teilpfade werden nur einmal berechnet, nur der jeweils beste Teilpfad zu einem bestimmten Punkt wird beibehalten
  - Genau dieses Vorgehen nennt man **Viterbi-Algorithmus**

# Viterbi-Algorithmus

- Paradigma: Speichern und wiederverwerten bereits berechneter Informationen
- Dazu bauen wir rekursiv eine Datenstruktur (*Trellis)* auf, die alle Pfade der Länge i enthält.
- Der Trellis besteht aus i *Schichten,* die jeweils alle N Zustände des HMM enthalten.
- Gibt es im HMM eine Kante von i nach j, dann gibt es in jeder Schicht S(t) eine Kante von i zum Knoten j in der Schicht S(t+1)

# Viterbi-Algorithmus

- Definiere Variable $\delta_t(i)$, die die Wahrscheinlichkeit der wahrscheinlichsten Folge von Zuständen ist, die in i endet und $O_1 \ldots O_t$ emittiert.
- Um den eigentlichen Pfad herauszufinden, speichern wir in $\psi_t(i)$ den letzten Zustand dieser Folge vor dem Zustand i.

Initialisierung :

$$\delta_1(i) = \pi_i * b_{i(O_1)} \qquad 1 \leq i \leq N$$

$$\psi_1(i) = 0$$

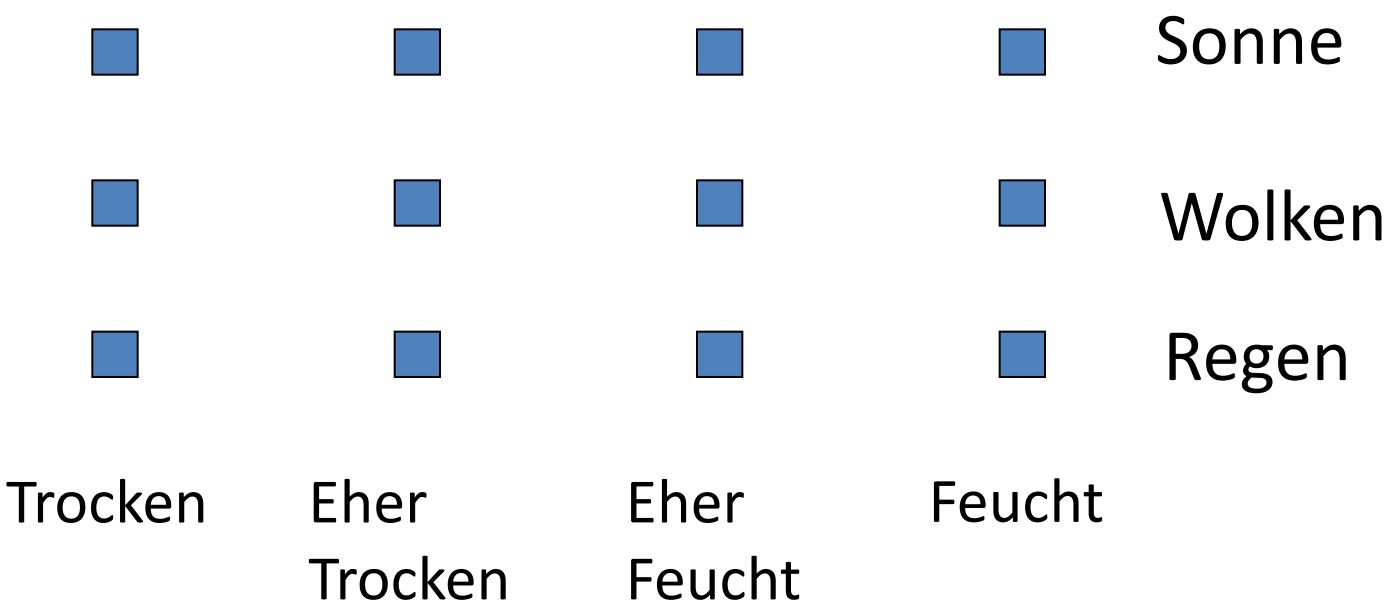
Rekursion :

$$\delta_t(j) = \max [\delta_{t-1}(i) a_{ij}] * b_{i(O_t)} \qquad 2 \leq t \leq T, \; 1 \leq i \leq N$$

$$\psi_t(j) = \operatorname{argmax} [\delta_{t-1}(i) a_{ij}] \qquad 2 \leq t \leq T, \; 1 \leq i \leq N$$

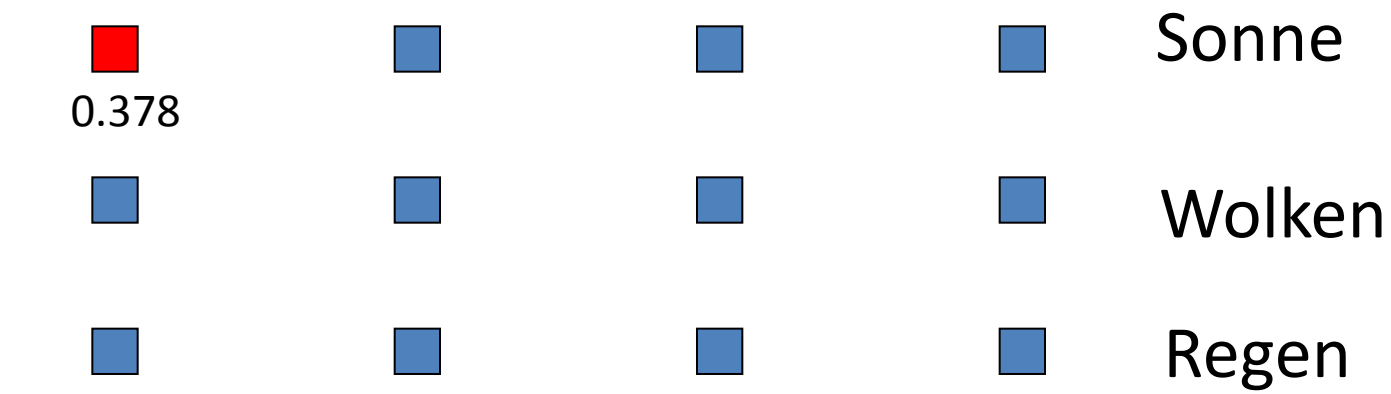
# Beispiel: Viterbi-Algorithmus

*Wie lässt sich zu der Beobachtung die Sequenz der wahrscheinlichsten verborgenen Zustände finden (Dekodierung)*



# Beispiel: Viterbi-Algorithmus

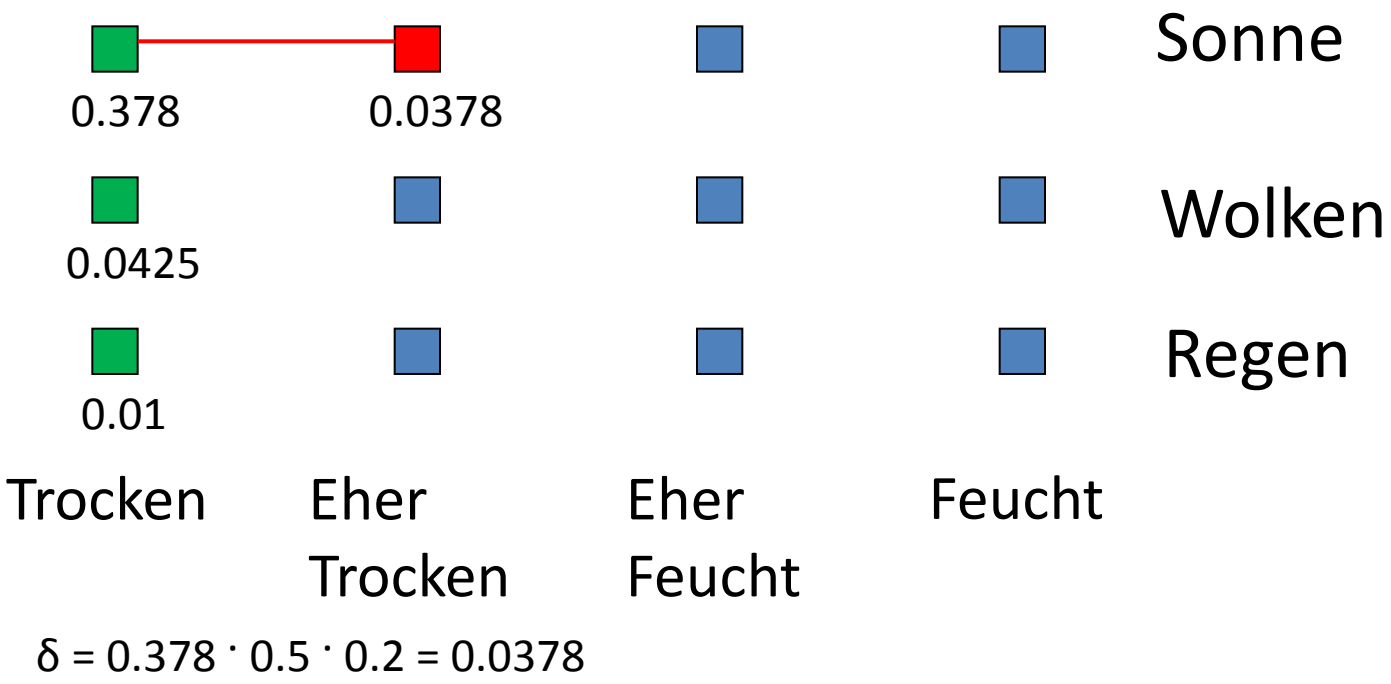
*Wie lässt sich zu der Beobachtung die Sequenz der wahrscheinlichsten verborgenen Zustände finden (Dekodierung)*




$\delta = 0.63 \cdot 0.6 = 0.378$

# Beispiel: Viterbi-Algorithmus

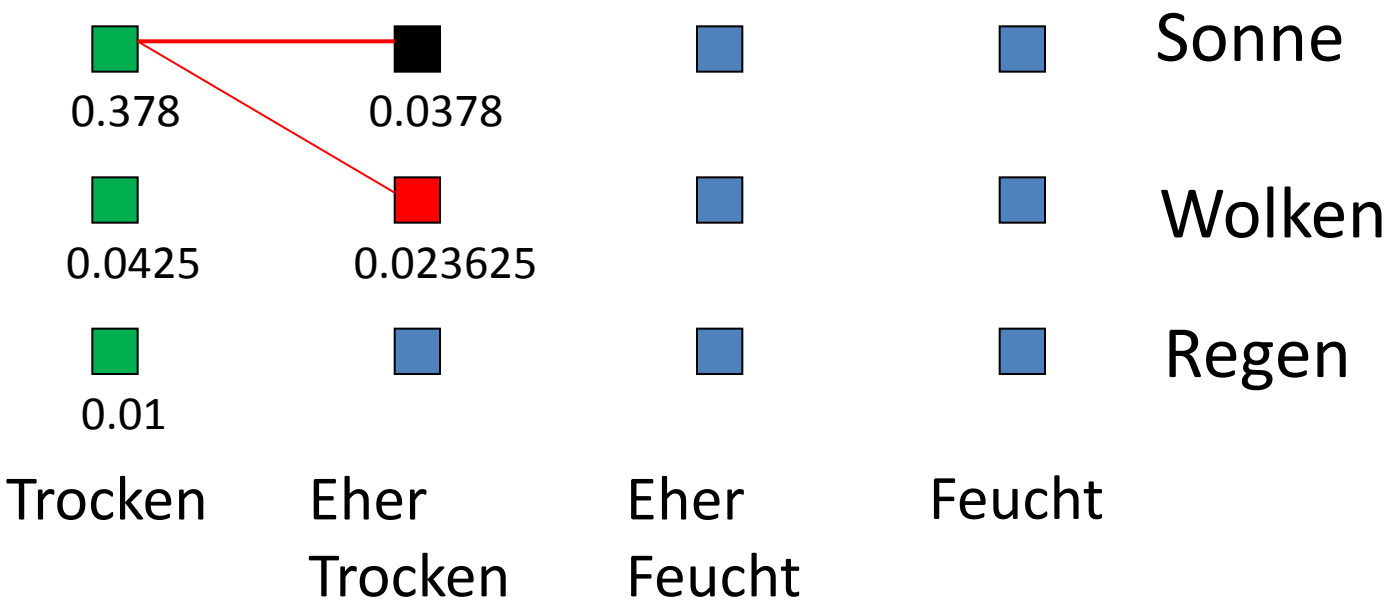
*Wie lässt sich zu der Beobachtung die Sequenz der wahrscheinlichsten verborgenen Zustände finden (Dekodierung)*




$\delta = 0.378 \cdot 0.5 \cdot 0.2 = 0.0378$

# Beispiel: Viterbi-Algorithmus

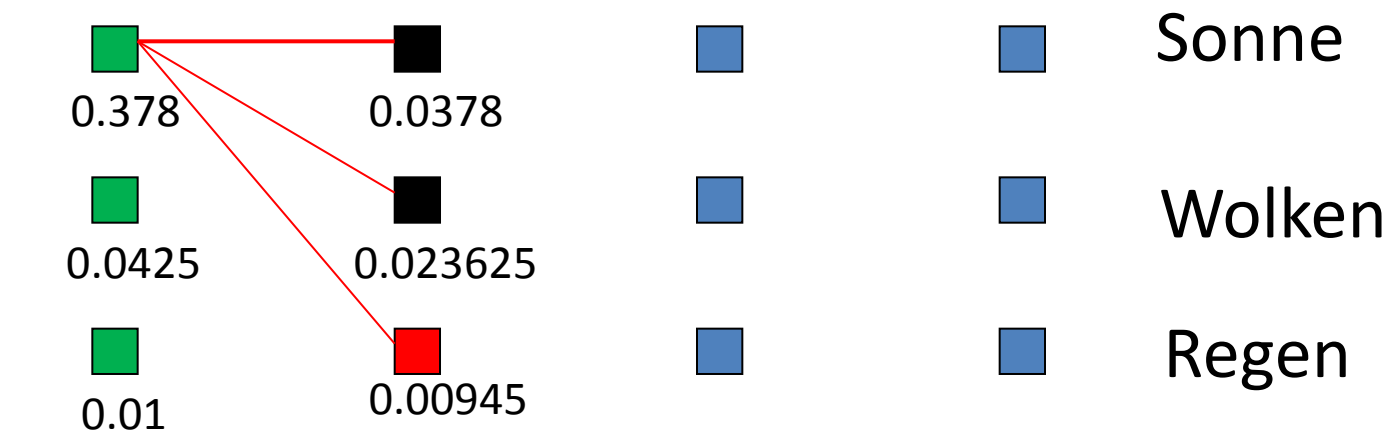
*Wie lässt sich zu der Beobachtung die Sequenz der wahrscheinlichsten verborgenen Zustände finden (Dekodierung)*




$\delta = 0.378 \cdot 0.25 \cdot 0.25 = 0.023625$

# Beispiel: Viterbi-Algorithmus

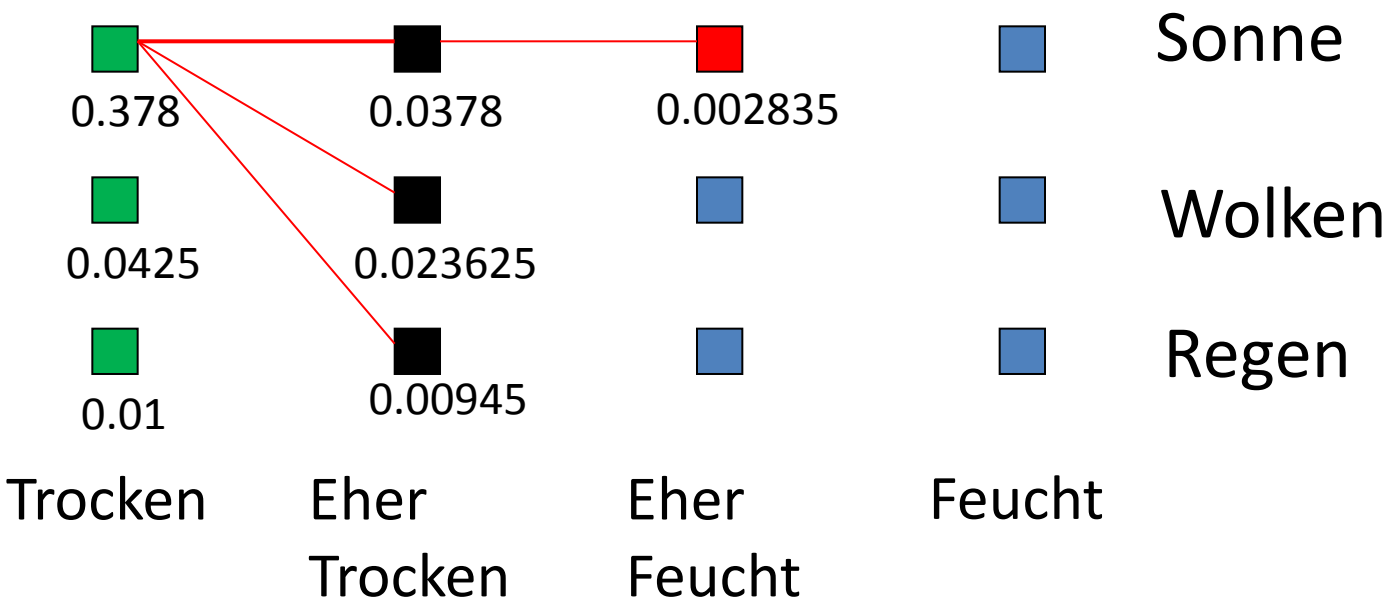
*Wie lässt sich zu der Beobachtung die Sequenz der wahrscheinlichsten verborgenen Zustände finden (Dekodierung)*




$\delta = 0.378 \cdot 0.25 \cdot 0.1 = 0.00945$

# Beispiel: Viterbi-Algorithmus

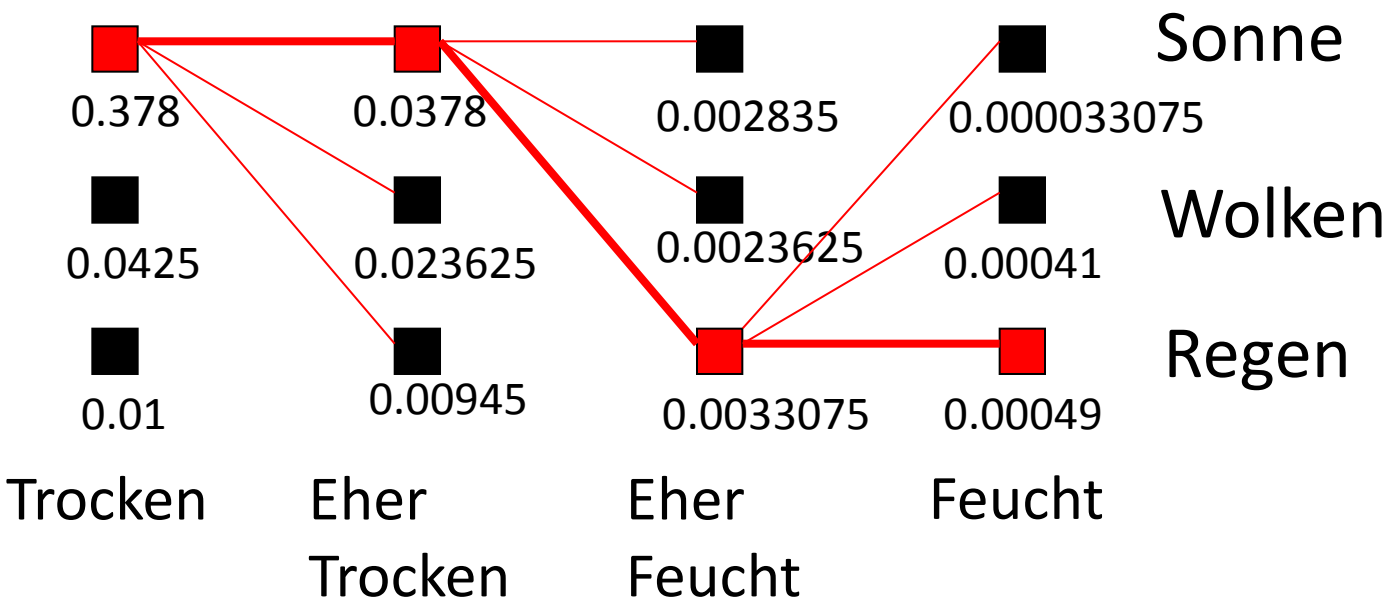
*Wie lässt sich zu der Beobachtung die Sequenz der wahrscheinlichsten verborgenen Zustände finden (Dekodierung)*




$\delta = 0.0378 \cdot 0.5 \cdot 0.15 = 0.002835$

# Beispiel: Viterbi-Algorithmus

*Wie lässt sich zu der Beobachtung die Sequenz der wahrscheinlichsten verborgenen Zustände finden (Dekodierung)*




Der rote Pfad zeigt den wahrscheinlichsten Weg.

# Zusammenfassung

- HMM: Chancen und Probleme

|  |  |
|---|---|
| • solide statistische Grundlage | • betrachten nur kleine Ausschnitte des Textes auf einmal |
| • weit verbreiteter Einsatz in unterschiedlichen Anwendungsgebieten | • Zusammenhänge weit auseinander stehender Textteile gehen verloren. |
| • effiziente Lernalgorithmen für das Trainieren von HMMs verfügbar | • ausreichend Trainingsdaten erforderlich |
| • HMM Implementierungen verfügbar:<br>- GHMM Library in C<br>- Jahmm in Java | • Label-Bias Problem→HMM "over-fitting" |

# Zusammenfassung

- Text Labeling: wichtiger preprocessing Schritt in der natürlichen Sprachverarbeitung
- HMM:
  - Doppelt stochastischer Prozess bestehend aus Markovkette und Beobachtungsfolge
  - eine der wichtigsten und gängigsten statistischen Prozessmodelle zum „Text Labeling“
  - Verbreiteter Einsatz im Bereich der Information Extraction (z.B. POS-Tagging, Chunking, NER)
- Neben HMMs existieren weitere Modelle, die innerhalb maschineller Lernverfahren für das „Text-Labeling“ Problem verwendet werden können: MEMM, CRF und hybride Modelle

## Danke für eure Aufmerksamkeit!

## Fragen?