UNIVERSITÄT LEIPZIG

NLP Lab Session 5
Hidden Markov Models

Felix Helfer
helfer@saw-leipzig.de

24.06.24 / 01.07.24

HIDDEN MARKOV MODELS

Kurzer Rückblick:

- HMMs als Verfahren für das **Sequence Labeling**
- Also: Modell, welches jeder **Einheit** in einer **Sequenz** ein **Label** zuordnet
- Anwendung z.B. bei **Part-of-Speech-Tagging**

(im Deutschen übrigens Markow)

MARKOV CHAINS

- HMM als Erweiterung einer **Markov Chain** (Markowkette).
- Markov Chains berechnen WKT für **Sequenzen beobachtbarer Zustände**.

Markov Chain

$Q = q_1\ q_2\ \ldots\ q_N$	*Menge von N Zuständen*
$A = a_{11}\ a_{12}\ \ldots\ a_{N1}\ \ldots\ a_{NN}$	*Übergangswahrscheinlichkeitsmatrix A*
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	*Initiale Wahrscheinlichkeitsverteilung über den Zuständen*

MARKOV CHAINS

- (Für Markowketten erster Ordnung) Enthält die **Markow-Annahme**. Für eine *Zustandssequenz* q_1, q_2, … q_i gilt:

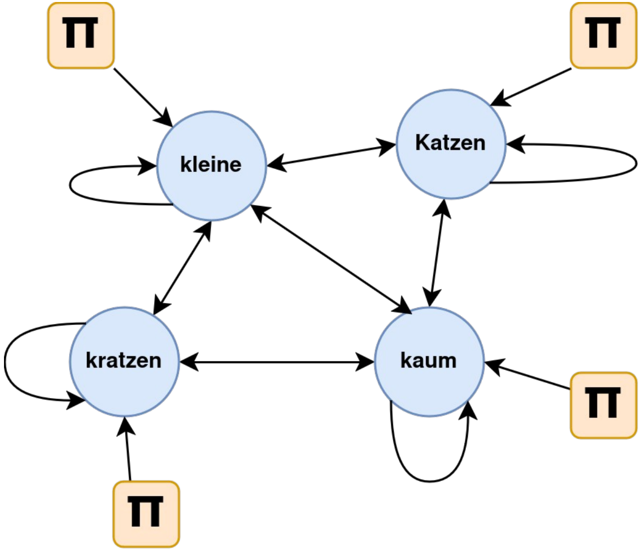
$$P(q_i = a | q_1 \ldots q_{i-1}) = P(q_i = a | q_{i-1})$$

Markov Chain

$Q = q_1\ q_2\ \ldots\ q_N$	*Menge von N Zuständen*
$A = a_{11}\ a_{12}\ \ldots\ a_{N1}\ \ldots\ a_{NN}$	*Übergangswahrscheinlichkeitsmatrix A*
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	*Initiale Wahrscheinlichkeitsverteilung über den Zuständen*

MARKOV CHAINS - BEISPIEL

Bigram-Modell



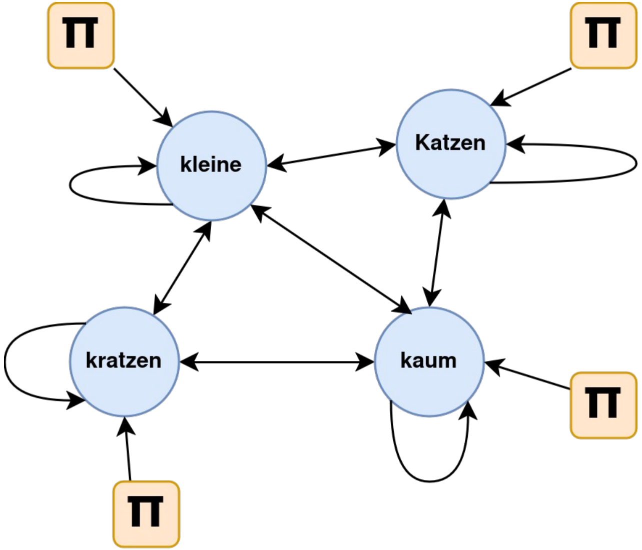
Übergangswahrscheinlichkeiten A

	kleine	Katzen	kratzen	kaum
π	0.6	0.2	0.1	0.1
kleine	0.0	0.8	0.1	0.1
Katzen	0.1	0.1	0.4	0.4
kratzen	0.3	0.4	0.0	0.3
kaum	0.3	0.4	0.3	0.0

→ *Wahrscheinlichkeit von „Kleine Katzen kratzen kaum“ vs. „kaum kratzen kleine Katzen“?*

MARKOV CHAINS - BEISPIEL

Bigram-Modell



Übergangswahrscheinlichkeiten A

	kleine	Katzen	kratzen	kaum
π	0.6	0.2	0.1	0.1
kleine	0.0	0.8	0.1	0.1
Katzen	0.1	0.1	0.4	0.4
kratzen	0.3	0.4	0.0	0.3
kaum	0.3	0.4	0.3	0.0

→ *Wahrscheinlichkeit von „Kleine Katzen kratzen kaum“ vs. „kaum kratzen kleine Katzen“?*

P(„kleine Katzen kratzen kaum“) = 0.6 * 0.8 * 0.4 * 0.3 = 0.0576
P(„kaum kratzen kleine Katzen“) = 0.1 * 0.3 * 0.3 * 0.8 = 0.0072

VON MARKOV CHAINS ZU HIDDEN MARKOV MODELS

Wieso sind Markov Chains nur bedingt für Aufgaben wie Part-of-Speech-Tagging geeignet?

VON MARKOV CHAINS ZU HIDDEN MARKOV MODELS

Wieso sind Markov Chains nur bedingt für Aufgaben wie Part-of-Speech-Tagging geeignet?

→ Die relevanten Zustände (also: POS-Tags) sind für gewöhnlich **versteckt** (d.h. nicht beobachtet).

Deshalb: Ein Modell für beobachtete *und* unbeobachtete Zustände (z.B. Token vs. POS-Tags) → **Hidden Markov Model**

HIDDEN MARKOV MODELS

Hidden Markov Model

$Q = q_1\ q_2\ \dots\ q_N$	*Menge von N Zuständen*
$A = a_{11}\ a_{12}\ \dots\ a_{N1}\ \dots\ a_{NN}$	*Übergangswahrscheinlichkeitsmatrix A*
$\pi = \pi_1, \pi_2, \dots, \pi_N$	*Initiale Wahrscheinlichkeitsverteilung über den Zuständen*
$O = o_1\ o_2\ \dots\ o_T$	*Sequenz von T Beobachtungen, gezogen aus Vokabular V*
$B = b_i(o_t)$	*Sequenz von Beobachtungs-/Emissionswahrschienlichkeiten*

HIDDEN MARKOV MODELS

- Es gilt (für HMMs erster Ordnung), ebenfalls die **Markow-Annahme** (für eine *Zustandssequenz* q_1, q_2, … q_i):

$$P(q_i = a | q_1 \dots q_{i-1}) = P(q_i = a | q_{i-1})$$

(WKT eines Zustands hängt nur ab vom vorigen Zustand q_{i-1})

- Des weiteren die **Ausgabeunabhängigkeit**:

$$P(o_i | q_1, \dots, q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i | q_i)$$

(WKT einer Beobachtung o_i hängt nur ab vom Zustand q_i der sie hervorgebracht hat)

HIDDEN MARKOV MODELS - BEISPIEL

Zeit für ein **Beispiel** – wir erstellen unseren eigenen **Bigramm-HMM-Tagger.**

→ ***Was benötigen wir dafür?***

HIDDEN MARKOV MODELS - BEISPIEL

Training

Unsere Trainingsdaten:

the/D fake/Ad cats/N hunt/V stupid/Ad mice/N

mice/N fake/V the/D hunt/N

the/D cats/N fake/V mice/N cats/N

N: Nomen
V: Verb
D: Determinativ
Ad: Adjektiv

HIDDEN MARKOV MODELS - BEISPIEL

Übergangswahrscheinlichkeiten A (inklusive Start)

Bsp:
P(V|N):
3 Übergänge N → V,
4 N insgesamt (mit Übergängen)
→ P(V|N) = 3/4

	N	V	D	Ad
π	0.33	0.0	0.67	0.0
N	0.25	**0.75**	0.0	0.0
V	0.33	0.0	0.33	0.33
D	0.67	0.0	0.0	0.33
Ad	1.0	0.0	0.0	0.0

HIDDEN MARKOV MODELS - BEISPIEL

Emissionswahrscheinlichkeiten B

Bsp:
P(cats|N):
3 Vorkommen N mit „cats“,
7 Vorkommen N insgesamt
→ P(cats|N) = 3/7

	the	fake	cats	hunt	stupid	mice
N	0.0	0.0	**0.43**	0.14	0.0	0.43
V	0.0	0.67	0.0	0.33	0.0	0.0
D	1.0	0.0	0.0	0.0	0.0	0.0
Ad	0.0	0.5	0.0	0.0	0.5	0.0

HIDDEN MARKOV MODELS - BEISPIEL

Training abgeschlossen.

Jetzt: **Beobachtungssequenz** als Input

cats hunt stupid homework

→ ***Was nun?***

HIDDEN MARKOV MODELS - BEISPIEL

Decoding

Input: **Beobachtungssequenz** $w_1 \ldots w_n$

Ziel: *wahrscheinlichste* **Tagsequenz** $t_1 \ldots t_n$

Also: $\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{argmax}\, P(t_1 \ldots t_n | w_1 \ldots w_n)$

→ ***Wie ist das machbar mit unserem HMM-Tagger?***

HIDDEN MARKOV MODELS - BEISPIEL

Decoding

(1) $$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{argmax}\, P\left(t_1 \dots t_n | w_1 \dots w_n\right)$$

HIDDEN MARKOV MODELS - BEISPIEL

Decoding

(1) $$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{argmax}\, P(t_1 \ldots t_n | w_1 \ldots w_n)$$

(2) $$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{argmax} \frac{P(w_1 \ldots w_n | t_1 \ldots t_n) P(t_1 \ldots t_n)}{P(w_1 \ldots w_n)}$$ (Bayes)

HIDDEN MARKOV MODELS - BEISPIEL

Decoding

(1) $$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{argmax}\, P(t_1 \dots t_n | w_1 \dots w_n)$$

(2) $$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{argmax}\, \frac{P(w_1 \dots w_n | t_1 \dots t_n)\, P(t_1 \dots t_n)}{P(w_1 \dots w_n)}$$ (Bayes)

(3) $$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{argmax}\, P(w_1 \dots w_n | t_1 \dots t_n)\, P(t_1 \dots t_n)$$ (Vereinfachung)

HIDDEN MARKOV MODELS - BEISPIEL

(3) $\hat{t}_{1:n} = \underset{t_1 \dots t_n}{argmax}\, P(w_1 \dots w_n | t_1 \dots t_n) P(t_1 \dots t_n)$

HIDDEN MARKOV MODELS - BEISPIEL

(3) $$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{argmax}\, P(w_1 \ldots w_n | t_1 \ldots t_n)\, P(t_1 \ldots t_n)$$

(4) $$P(t_1 \ldots t_n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1})$$ (Markow-Annahme)

HIDDEN MARKOV MODELS - BEISPIEL

(3) $$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{argmax}\, P(w_1 \dots w_n | t_1 \dots t_n) P(t_1 \dots t_n)$$

(4) $$P(t_1 \dots t_n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1})$$ (Markow-Annahme)

(5) $$P(w_1 \dots w_n | t_1 \dots t_n) \approx \prod_{i=1}^{n} P(w_i | t_i)$$ (Ausgabeunabhängigkeit)

HIDDEN MARKOV MODELS - BEISPIEL

(3) $\hat{t}_{1:n} = \underset{t_1 \dots t_n}{argmax}\, P(w_1 \dots w_n | t_1 \dots t_n)\, P(t_1 \dots t_n)$

(4) $P(t_1 \dots t_n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1})$ (Markow-Annahme)

(5) $P(w_1 \dots w_n | t_1 \dots t_n) \approx \prod_{i=1}^{n} P(w_i | t_i)$ (Ausgabeunabhängigkeit)

(6) $\hat{t}_{1:n} = \underset{t_1 \dots t_n}{argmax}\, P(t_1 \dots t_n | w_1 \dots w_n) \approx \underset{t_1 \dots t_n}{argmax} \prod_{i=1}^{n} \overbrace{P(w_i | t_i)}^{\text{emission}}\, \overbrace{P(t_i | t_{i-1})}^{\text{transition}}$

HIDDEN MARKOV MODELS - BEISPIEL

Decoding

Input: *cats hunt stupid homework*

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{argmax}\, P(t_1 \ldots t_n | w_1 \ldots w_n) \approx \underset{t_1 \ldots t_n}{argmax} \prod_{i=1}^{n} \overbrace{P(w_i|t_i)}^{\text{emission}}\, \overbrace{P(t_i|t_{i-1})}^{\text{transition}}$$

→ ***Wie finden wir (auch für größere Tagsets) schnell eine Lösung?***

HIDDEN MARKOV MODELS - BEISPIEL

Decoding

Viterbi-Algorithmus

N: Nomen
V: Verb
D: Determinativ
Ad: Adjektiv

A:

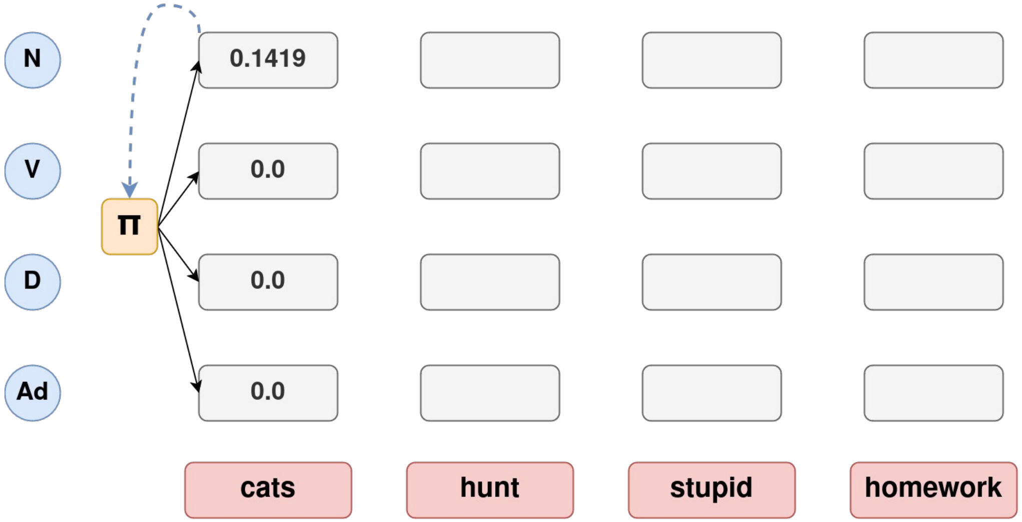
	N	V	D	Ad
π	0.33	0.0	0.67	0.0
N	0.25	0.65	0.0	0.1
V	0.33	0.0	0.33	0.33
D	0.67	0.0	0.0	0.33
Ad	1.0	0.0	0.0	0.0

B:

	the	fake	cats	hunt	stupid	mice
N	0.0	0.0	0.43	0.14	0.0	0.43
V	0.0	0.67	0.0	0.33	0.0	0.0
D	1.0	0.0	0.0	0.0	0.0	0.0
Ad	0.0	0.5	0.0	0.0	0.5	0.0

HIDDEN MARKOV MODELS - BEISPIEL

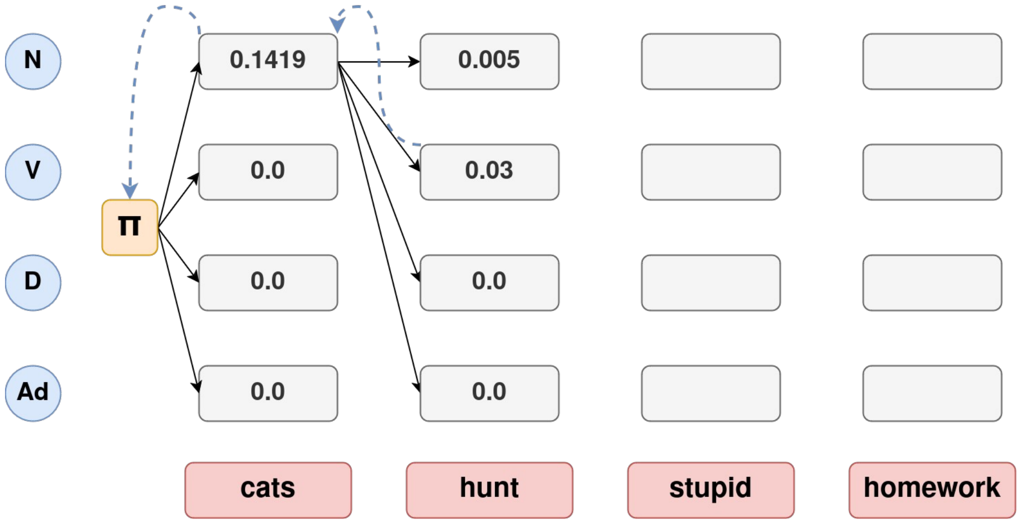
Decoding - Viterbi-Algorithmus



*P(N | π) * P(cats | N) = 0.1419*

HIDDEN MARKOV MODELS - BEISPIEL

Decoding - Viterbi-Algorithmus

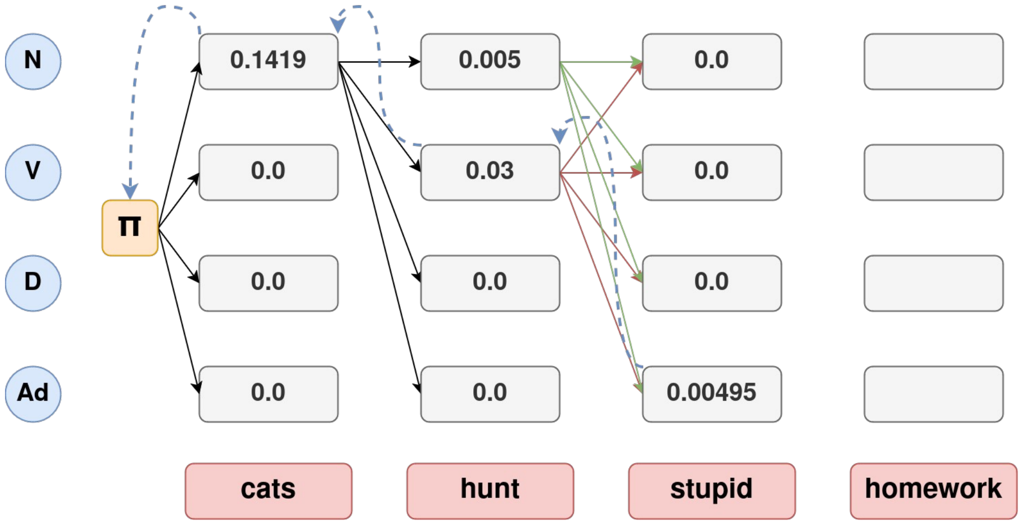


*0.1419 * P(N | N) * P(hunt | N) = 0.005*

*0.1419 * P(V | N) * P(hunt | V) = 0.03*

HIDDEN MARKOV MODELS - BEISPIEL

Decoding - Viterbi-Algorithmus



Hier: zwei Optionen für unterste Zelle → **Maximum wählen**!

*max(0.005 * P(Ad | N) * P(stupid | Ad) = 0.00025*

*0.03 * P(Ad | V) * P(stupid | Ad) = 0.00495)*

HIDDEN MARKOV MODELS - BEISPIEL

Decoding - Viterbi-Algorithmus

cats hunt stupid homework

→ *Was tun bei **Out-of-Vocabulary-Worten**?*

HIDDEN MARKOV MODELS - BEISPIEL

Decoding - Viterbi-Algorithmus

cats hunt stupid homework

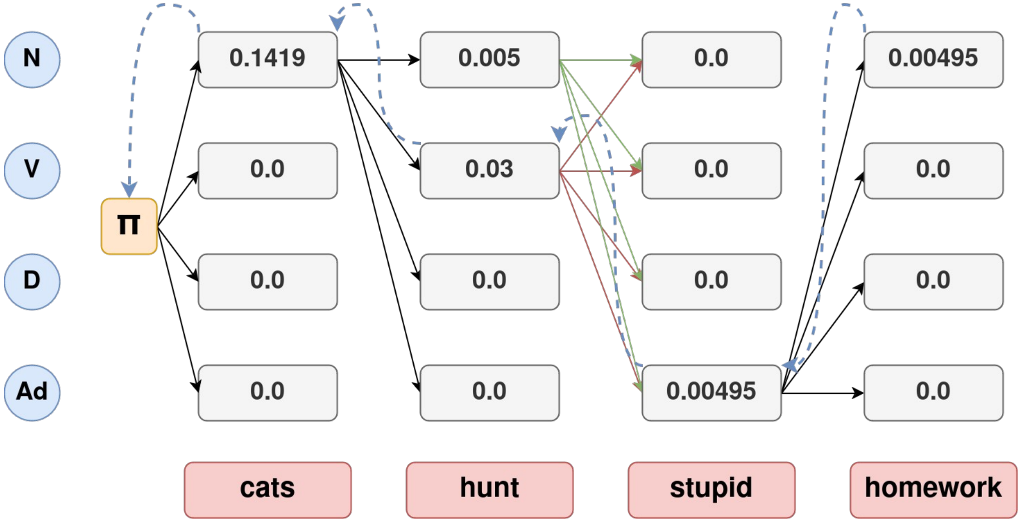
→ *Was tun bei **Out-of-Vocabulary-Worten**?*

Zum Beispiel:

- Tag-Häufigkeiten von Worten mit Häufigkeit 1 heranziehen
- Morphologie o.äh. berücksichtigen
- Nur $P(t_i|t_{i-1})$ beachten (*hier verwendet*)

HIDDEN MARKOV MODELS - BEISPIEL

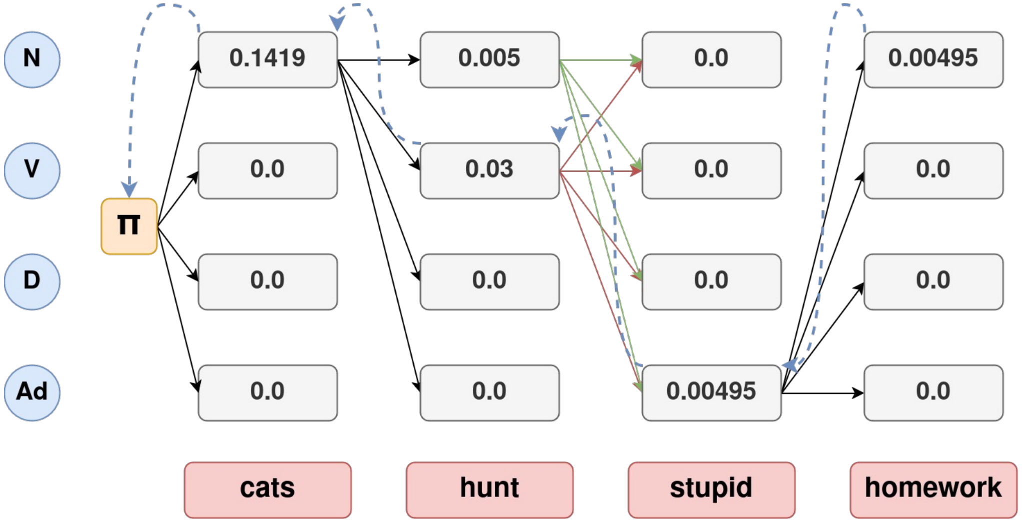
Decoding - Viterbi-Algorithmus



*0.00495 * P(N | Ad) = 0.00495*

HIDDEN MARKOV MODELS - BEISPIEL

Decoding - Viterbi-Algorithmus



Zuletzt: **Backtracing** des optimalen Pfades ($N \rightarrow Ad \rightarrow V \rightarrow N$ für „*N V Ad N*“)

ZUSAMMENFASSUNG

Heute besprochen:

- Markov Chains
- Hidden Markov Models
- Viterbi-Algorithmus

Danke fürs Zuhören!

Quelle:

D. Jurafsky, J. H. Martin: Speech and Language Processing (3rd ed. Draft), https://web.stanford.edu/~jurafsky/slp3/, 2021