Einführung in die Computerlinguistik

HMM POS-Tagging

Laura Kallmeyer

Heinrich-Heine-Universität Düsseldorf

Summer 2019

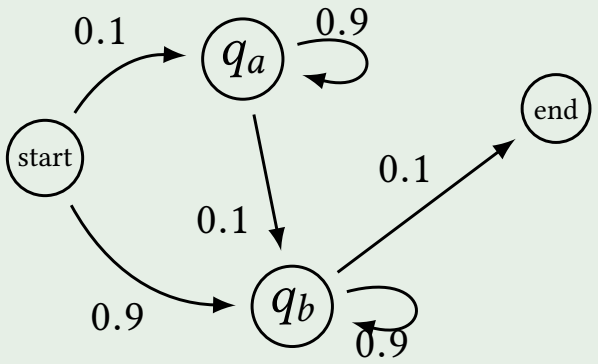
Motivation

Hidden Markov Models (**HMMs**)

- are weighted finite state automata
- with states emitting outputs.
- Transitions and emissions are weighted.

Motivation

HMM



Emission probabilities:
in q_a, a is emitted with prob. 0.9;
in q_a, b is emitted with prob. 0.1;
in q_b, b is emitted with prob. 0.9;
in q_b, a is emitted with prob. 0.1;

Given a sequence bb, what is the most probable path traversed by the automaton, resulting in this output?

The states in the automaton correspond to classes, i.e., the search for the most probable path amounts to the search for the most probable class sequence.

Example

POS Tagging: the classes are POS tags, the emissions are words

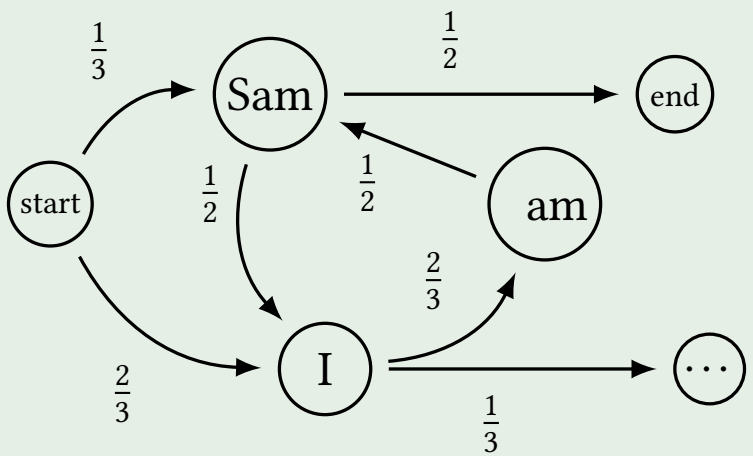
Hidden Markov Models

A **Markov chain** is a weighted automaton in which

- weights are probabilities, i.e., all weights are between 0 and 1 and the sum of the weights of all outgoing edges of a state is 1, and
- the input sequence uniquely determines the states the automaton goes through.

A Markov chain is actually a bigram language model.

Markov Chain for LM example slide 9



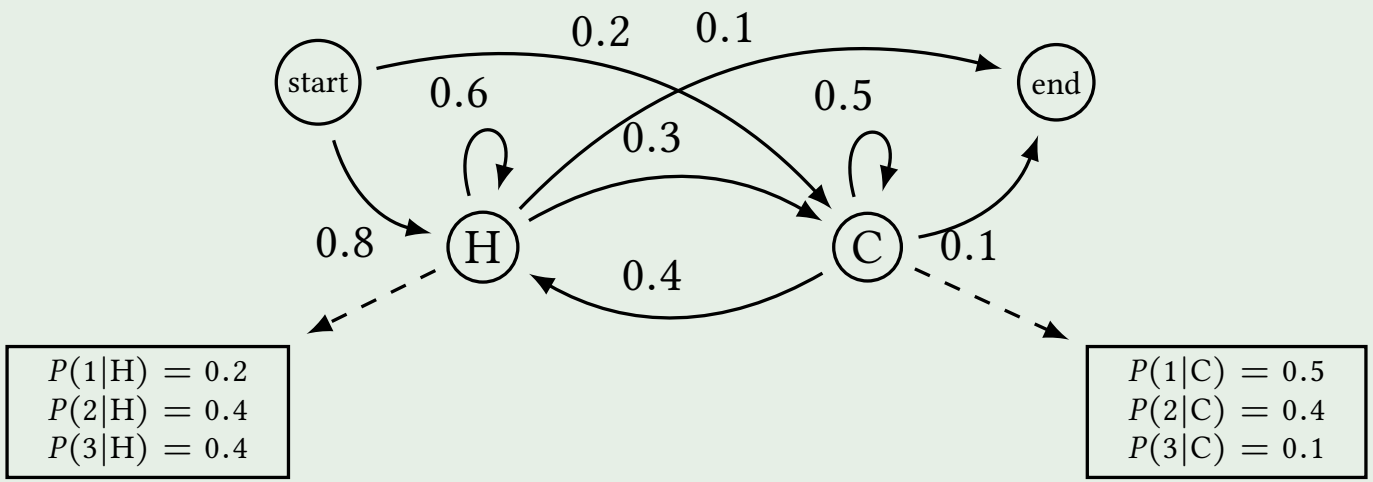
Hidden Markov Models

Markov chains are useful when we want to compute the probability for a sequence of events that we can observe.

Hidden Markov Models compute probabilities for a sequence of hidden events, given a sequence of observed events.

Example from Jurafsky and Martin (2017), after Eisner (2002)

HMM for inferring weather (hot = H, cold = C) from numbers of ice creams eaten by Jason.



Hidden Markov Models

HMM

A HMM is a tuple $\langle Q, A, O, B, q_0, q_F \rangle$, with

- $Q = \{q_1, \ldots, q_N\}$ a finite set of states;
- A a $|Q| \times |Q|$ matrix, the transition probability matrix, with $0 \leq a_{ij} \leq 1$ for all $1 \leq i \leq |Q|, 1 \leq j \leq |Q|$.
- O a finite set of observations;
- a sequence B of $|Q|$ functions $b_i : O \to \mathbb{R}$, the emission probabilities with $\sum_{o \in O} b_i(o) = 1$ for all $1 \leq i \leq |Q|$
- $q_0, q_F \notin Q$ are special start and end states, associated with transition probabilities a_{0i} and a_{iF} ($0 \leq a_{0i}, a_{iF} \leq 1$) for all $1 \leq i \leq |Q|$.

For all i, $0 \leq i \leq |Q|$, it holds that $\sum_{j=1}^{|Q|} a_{ij} + a_{iF} = 1$.

Hidden Markov Models

Given an HMM, we can compute the following:

- The probability of a specific sequence of states $\mathbf{q} = q_{i_1} \dots q_{i_n}$:

$$P(q_{i_1} \dots q_{i_n}) = a_{0i_1} \cdot a_{i_nF} \cdot \prod_{j=1}^{n-1} a_{i_j i_{j+1}}$$

- The probability of a combination of a sequence of states $\mathbf{q} = q_{i_1} \dots q_{i_n}$ and observations $\mathbf{o} = o_1 \dots o_n$:

$$P(q_{i_1} \dots q_{i_n}, o_1 \dots o_n) = a_{0i_1} \cdot a_{i_nF} \cdot b_{i_1}(o_1) \prod_{j=1}^{n-1} a_{i_j i_{j+1}} b_{i_{j+1}}(o_{j+1})$$

Example from slide 6

$P(HH) = 0.8 \cdot 0.6 \cdot 0.1$

$P(CH, 12) = 0.2 \cdot 0.4 \cdot 0.1 \cdot 0.5 \cdot 0.4$

Hidden Markov Models

- The probability of a sequence of observations $\mathbf{o} = o_1 \ldots o_n$:

$$P(o_1 \ldots o_n) = \sum_{q_{i_1} \ldots q_{i_n} \in Q^n} (a_{0 i_1} \cdot a_{i_n F} \cdot b_{i_1}(o_1) \prod_{j=1}^{n-1} a_{i_j i_{j+1}} b_{i_{j+1}}(o_{j+1}))$$

- The most probable sequence of states given a sequence of observations $\mathbf{o} = o_1 \ldots o_n$:

$$\arg\max_{q_{i_1} \ldots q_{i_n} \in Q^n} (a_{0 i_1} \cdot a_{i_n F} \cdot b_{i_1}(o_1) \prod_{j=1}^{n-1} a_{i_j i_{j+1}} b_{i_{j+1}}(o_{j+1}))$$

Example from slide 6

$$\begin{aligned} P(13) \quad = \quad & 0.8 \cdot 0.6 \cdot 0.1 \cdot 0.2 \cdot 0.4 && \text{(HH)} \\ & +0.8 \cdot 0.3 \cdot 0.1 \cdot 0.2 \cdot 0.1 && \text{(HC)} \\ & +0.2 \cdot 0.5 \cdot 0.1 \cdot 0.5 \cdot 0.1 && \text{(CC)} \\ & +0.2 \cdot 0.4 \cdot 0.1 \cdot 0.5 \cdot 0.4 && \text{(CH)} \end{aligned}$$

Best sequence of states for 13: *HH*

POS Tags (1)

Jurafsky and Martin (2009)
POS = part-of-speech Tags sind morphosyntaktische Kategorien von Wortformen.
Bsp. (Brown Corpus):

```
(The AT) (Fulton NP-TL) (County NN-TL)
(Grand JJ-TL) (Jury NN-TL) (said VBD)
(Friday NR) (an AT) (investigation NN) (of
IN) (Atlanta's NP) (recent JJ) (primary NN)
(election NN) (produced VBD) (“ “) (no AT)
(evidence NN) (” ”) (that CS) (any DTI)
(irregularities NNS) (took VBD) (place NN)
(. .)
```

POS Tags (2)

Bsp. (Penn Treebank):

```
(Pierre NNP) (Vinken NNP) (, ,) (61 CD)
(years NNS) (old JJ) (, ,) (will MD) (join,
VB) (the DT) (board NN) (as IN) (a DT)
(nonexecutive JJ) (director NN) (Nov. NNP)
(29 CD) (. .)
```

POS Tags (3)

Es gibt kein fest vorgegebenes Inventar von möglichen POS Tags. Die Wahl der POS Tags hängt von der jeweiligen Sprache und von der Anwendung ab, die mit dem getaggten Text vorgenommen werden soll. Morphologisch reichere Sprachen haben in der Regel größere Tagsets.

- Penn Treebank Tagset (PTB, American English): 45 POS-Tags
- Brown Corpus (American English): 87 POS-Tags
- British National Corpus (BNC, British English) basic tagset: 61 POS-Tags
- Stuttgart-Tübingen Tagset (STTS) für das Deutsche: 54 POS-Tags.
- Prague Dependency Treebank (PDT, Tschechisch): 4288 POS-Tags.

HMM POS Tagging (1)

Problem: Gegeben eine Folge w_1^n von n Wörtern, wollen wir die wahrscheinlichste Folge $\hat{t}_1^n$ aller möglichen Folgen t_1^n von n POS Tags für diese Wortfolge ermitteln.

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

$\arg\max_x f(x)$ bedeutet "das x, für das $f(x)$ maximal groß wird".

Wenn wir dies als HMM modellieren wollen, in dem die POS Tags die versteckten Zustände sind, müssen wir diese als gegeben annehmen, also $P(t_1^n | w_1^n)$ durch $P(w_1^n | t_1^n)$ ausdrücken.

Formel von Bayes

Da $P(A \cap B) = P(A|B)P(B)$ und $P(B \cap A) = P(B|A)(P(A)$, gilt $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

HMM POS Tagging (2)

Angewendet auf unser Problem: $P(t_1^n|w_1^n) = \frac{P(w_1^n|t_1^n)P(t_1^n)}{P(w_1^n)}$

Ermitteln des besten POS-Tag Sequenz:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n|w_1^n) = \arg\max_{t_1^n} \frac{P(w_1^n|t_1^n)P(t_1^n)}{P(w_1^n)}$$

Der Faktor $\frac{1}{P(w_1^n)}$ ist für alle Tag Sequenzen gleich, kann also weggelassen werden. Daraus folgt

$$\hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n|t_1^n)P(t_1^n)$$

HMM POS Tagging (3)

$\hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n|t_1^n)P(t_1^n)$ ist immer noch zu aufwändig zu berechnen.

Annäherungen:

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

Wenn wir Trainingsdaten in Form eines getaggten Korpus zur Verfügung haben, können wir die benötigten Wahrscheinlichkeiten durch Abzählen ermitteln. $C(x)$ sei die Anzahl von Auftreten von x im Korpus.

HMM POS Tagging (4)

Maximum Likelihood Estimates (MLE)

- Tagübergangswahrscheinlichkeit:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

Bsp. Brown Corpus: $P(NN|DT) = \frac{C(DT,NN)}{C(DT)} = \frac{56509}{116454} = 0.49$

- Wortwahrscheinlichkeit:

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

Bsp. Brown Corpus: $P(is|VBZ) = \frac{C(VBZ,is)}{C(VBZ)} = \frac{10073}{21627} = 0.47$

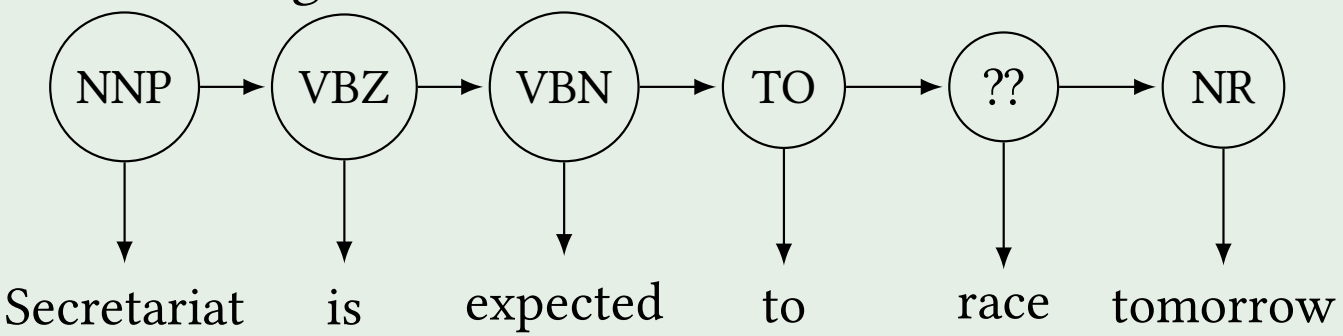
HMM POS Tagging (5)

Ambiguity between two POS tags

race, Ambiguität zwischen VB und NN.

(1) Secretariat is expected to race tomorrow
NNP VBZ VBN TO ?? NR
(Brown tagset)

Entscheidung, ob VB oder NN:



HMM POS Tagging (6)

Example continued

Übergangs- und Wortwahrscheinlichkeiten, die unterschiedlich sind:

$P(\text{NN}|\text{TO}) = 0.00047$ $\quad$ $P(\text{VB}|\text{TO}) = 0.83$

$P(\text{race}|\text{NN}) = 0.00057$ $\quad$ $P(\text{race}|\text{VB}) = 0.00012$

$P(\text{NR}|\text{NN}) = 0.0012$ $\quad$ $P(\text{NR}|\text{VB}) = 0.0027$

Je nach POS Tag für *race* unterscheiden sich die Wahrscheinlichkeiten der beiden Folgen in dem Faktor

NN: $\quad P(\text{NN}|\text{TO})P(\text{race}|\text{NN})P(\text{NR}|\text{NN}) = 0.00000000032$

VB: $\quad P(\text{VB}|\text{TO})P(\text{race}|\text{VB})P(\text{NR}|\text{VB}) = 0.00000027$

Bigram-HMMs (1)

Formalisierung als gewichteter endlicher Automat. Der Automat erlaubt uns, über beobachtete Ereignisse (die zu taggenden Wörter) zu sprechen und über versteckte Ereignisse (Folgen von POS Tags). Daher die Bezeichnung Hidden Markov model (HMM).
Wir nehmen an, dass die Zustände numeriert sind, $Q = \{t_1, \ldots, t_{|Q|}\}$. (D.h. t_i ist POS Tag Nummer i, nicht i-tes POS Tag in der Reihenfolge der getaggten Eingabe.)

Ein HMM besteht aus

- Zuständen (= POS Tags $Q = \{t_1, \ldots, t_{|Q|}\}$),
- Übergangswahrscheinlichkeiten $a_{i,j}$ (= $P(t_j|t_i)$)
- und Emissionswahrscheinlichkeiten $b_i(w)$ (= $P(w|t_i)$).

(Siehe Folie 6.)

Bigram-HMMs (2)

HMM POS Tagger

$Q = \{Det, N, Adj, V\}$, $O = \{the, chief, rules, \ldots\}$ und folgende Wahrscheinlichkeiten:

- Emissionswahrscheinlichkeiten ($b_i(w) = P(w|q_i)$):
  $P(\text{the}|\text{Det}) = 1$ $P(\text{chief}|\text{N}) = 3 \cdot 10^{-3}$ $P(\text{rules}|\text{N}) = 5 \cdot 10^{-3}$
  $P(\text{chief}|\text{Adj}) = 4 \cdot 10^{-3}$ $P(\text{rules}|\text{V}) = 6 \cdot 10^{-3}$
  Alle anderen Emissionswahrscheinlichkeiten für *chief*, *the* und *rules* seien 0.

- Teil der Übergangswahrscheinlichkeiten ($A_{i,j} = P(q_j|q_i)$):
  $P(\text{N}|\text{Det}) = 5 \cdot 10^{-1}$ $P(\text{N}|\text{N}) = 1 \cdot 10^{-1}$ $P(\text{N}|\text{Adj}) = 5 \cdot 10^{-1}$
  $P(\text{Adj}|\text{Det}) = 3 \cdot 10^{-1}$ $P(\text{V}|\text{N}) = 4 \cdot 10^{-1}$ $P(\text{V}|\text{Adj}) = 1 \cdot 10^{-1}$

  Die Wahrscheinlichkeit, dass ein Det am Satzanfang steht, ist 1, die, dass auf ein N oder V ein Satzende folgt, ist jeweils $0.1 = 1 \cdot 10^{-1}$.

Bigram-HMMs (3)

Ziel: Gegeben ein HMM und eine Folge von Beobachtungen $\mathbf{o} = \mathbf{o}_1 \ldots \mathbf{o}_n$ soll ermittelt werden, welche Folge von Zuständen die Beobachtung am wahrscheinlichsten macht. Dies nennt man auch **decoding**.

Viterbi Algorithmus:

- Wir durchlaufen die Beobachtungen von links nach rechts und füllen dabei eine Matrix *viterbi* von Pfadwahrscheinlichkeiten in Kombination mit Emissionen.
- *viterbi* ist eine $(n \times (|Q| + 1))$-Matrix. Das Feld $viterbi[i, j]$ soll die Wahrscheinlichkeit des besten Pfads der Länge i mit letzten Zustand t_j in Kombination mit der Emission von $w_1 \ldots w_i$ angeben. Anders gesagt, die Wahrscheinlichkeit der besten Möglichkeit, nach Abarbeiten von $w_1 \ldots w_i$ in t_j zu landen.
- q_F bekommt den Index $|Q| + 1$.
- Außerdem führen wir Backpointer mit, die uns für $viterbi[i, j]$ sagen, welcher Zustand bei diesem besten Pfad t_j voranging.

Bigram-HMMs (4)

We fill a $n \times (|Q| + 1)$ matrix v such that

$$v_{i,j} = \max_{\mathbf{q}_1 \cdots \mathbf{q}_{i-1} \in Q^{i-1}} P(\mathbf{o}_1 \ldots \mathbf{o}_i, \mathbf{q}_1 \ldots \mathbf{q}_{i-1}, \mathbf{q}_i = t_j)$$

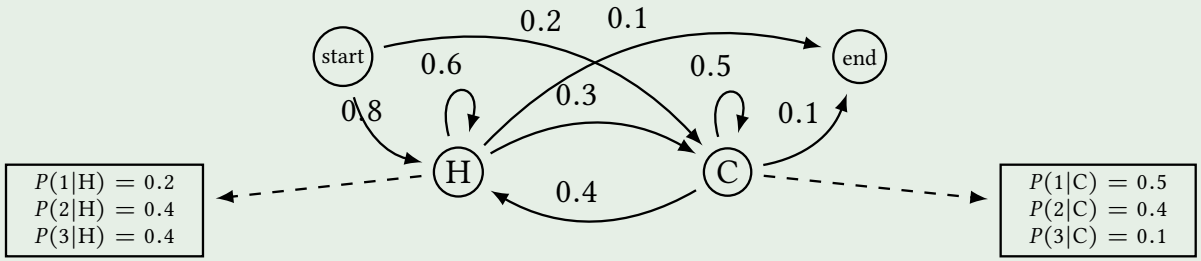
Viterbi algorithm

1. $v_{1,j} = a_{0j} b_j(\mathbf{o}_1)$ for $1 \leq j \leq |Q|$
2. $v_{i,j} = \max_{1 \leq k \leq |Q|} v_{i-1,k} a_{kj} b_j(\mathbf{o}_i)$ for $1 \leq i \leq n$ and $1 \leq j \leq |Q|$
3. $v_{n,|Q|+1} = \max_{\mathbf{q} \in Q^n} P(\mathbf{o}, \mathbf{q}) = \max_{1 \leq k \leq |Q|} v_{n,k} a_{kF}$

In addition, in order to retreive the best state sequence, each entry $v_{i,j}$ is equipped with a backpointer to the state that has lead to the maximum.

Bigram-HMMs (5)

Ice cream weather example continued



output sequence 313:

H	$32 \cdot 10^{-2}$, start	$384 \cdot 10^{-4}$, H	$9216 \cdot 10^{-6}$, H
C	$2 \cdot 10^{-2}$, start	$480 \cdot 10^{-4}$, H	$24 \cdot 10^{-4}$, C
	3	1	3

$v_{1j} = a_{0j} b_j(\mathbf{o}_1)$ for $1 \leq j \leq |Q|$
$v_{ij} = \max_{1 \leq k \leq |Q|} v_{i-1k} a_{kj} b_j(\mathbf{o}_i)$ for $1 \leq i \leq n$ and $1 \leq j \leq |Q|$
$\max_{\mathbf{q} \in Q^n} P(\mathbf{o}, \mathbf{q}) = \max_{1 \leq k \leq |Q|} v_{nk} a_{kF}$

most probable weather sequence is HHH with probability $9216 \cdot 10^{-7}$

Bigram-HMMs (6)

Im Fall von POS Tagging sind die Tags die Zustände, es wird also die wahrscheinlichste Folge von Tags für eine gegebene Eingabe ermittelt.

HMM for POS Tagging, Viterbi

q_F				$360 \cdot 10^{-9}$, V
N		$15 \cdot 10^{-4}$, Det	$300 \cdot 10^{-8}$, Adj	
V			$360 \cdot 10^{-8}$, N	
Adj		$12 \cdot 10^{-4}$, Det		
Det	1, q_0			
	1: the	2: chief	3: rules	

chief, Adj: $1 \cdot 3 \cdot 10^{-1} \cdot 4 \cdot 10^{-3}$ *chief*, N: $1 \cdot 5 \cdot 10^{-1} \cdot 3 \cdot 10^{-3}$

rules, N: $\max\{12 \cdot 10^{-4} \cdot 5 \cdot 10^{-1} \cdot 5 \cdot 10^{-3}$ Vorgänger Adj , $15 \cdot 10^{-4} \cdot 1 \cdot 10^{-1} \cdot 5 \cdot 10^{-3}$ Vorgänger N$\}$

rules, V: $\max\{12 \cdot 10^{-4} \cdot 1 \cdot 10^{-1} \cdot 6 \cdot 10^{-3}$ Vorgänger Adj , $15 \cdot 10^{-4} \cdot 4 \cdot 10^{-1} \cdot 6 \cdot 10^{-3}$ Vorgänger N$\}$

Die beste POS-TAG Folge ist demnach Det N V.

Bigram-HMMs (7)

Pseudo-Code for Viterbi

`for` q `in range(`$|Q|$`):`

$$viterbi(1, q) = a_{0,q} \cdot b_q(o_1)$$
$$backpointer(1, q) = 0$$

`for` i `from 2 to` n`:`

`for all` $q \in Q$`:`

$$viterbi(i, q) = \max_{q' \in |Q|}(viterbi(i-1, q') \cdot a_{q',q} \cdot b_q(o_i))$$
$$backpointer(i, q) = \arg\max_{q' \in |Q|}(viterbi(i-1, q') \cdot a_{q',q})$$

$$viterbi(n, q_F) = \max_{q' \in |Q|}(viterbi(n, q') \cdot a_{q',q_F})$$

$$backpointer(n, q_F) = \arg\max_{q' \in |Q|}(viterbi(n, q') \cdot a_{q',q_F})$$

Bigram-HMMs (8)

- Anschließend erhält man die beste Tagsequenz (in umgekehrter Reihenfolge), wenn man, von $backpointer(q_f, n)$ ausgehend den Backpointern folgt.
- Die Wahrscheinlichkeit dieser Tagsequenz steht in $viterbi(q_f, n)$.

HMM: best Tag sequence

q_F				$360 \cdot 10^{-9}$, V
N		$15 \cdot 10^{-4}$, Det	$300 \cdot 10^{-8}$, Adj	
V			$360 \cdot 10^{-8}$, N	
Adj		$12 \cdot 10^{-4}$, Det		
Det	1, q_0			
	1: the	2: chief	3: rules	

Beste POS Tag Sequenz ist Det N V, Wahrscheinlichkeit $36 \cdot 10^{-8}$.

Trigram-HMMs (1)

Annahme bisher: Die Übergangswahrscheinlichkeit dafür, dass ein bestimmtes Tag als nächstes folgt, ist nur von dem vorhergehenden Tag abhängig.

Erweiterung: es werden die beiden vorhergehenden Tags berücksichtigt:

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1}, t_{i-2})$$

Zusätzlich wird noch das Satzende berücksichtig, genauer die Wahrscheinlichkeit, dass das letzte Tag am Satzende steht (t_{n+1} ist ein neu eingeführtes Satzendetag):

$$\begin{aligned} \hat{t}_1^n &= \arg\max_{t_1^n} P(w_1^n | t_1^n) P(t_1^n) \\ &= \arg\max_{t_1^n} \left[\prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1}, t_{i-2})\right] P(t_{n+1} | t_n) \end{aligned}$$

Trigram-HMMs (2)

Problem: Man hat oft nicht genug Daten, um alle Wahrscheinlichkeiten abschätzen zu können:

$$P(t_i|t_{i-1}, t_{i-2}) = \frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})}$$

Viele dieser Anzahlen werden 0 sein, da man nicht alle Folgen im Trainingskorpus findet.

Lösung: Kombination von Tri-, Bi- und Unigrams:

$$P(t_i|t_{i-1}, t_{i-2}) = \lambda_3 \hat{P}(t_i|t_{i-1}, t_{i-2}) + \lambda_2 \hat{P}(t_i|t_{i-1}) + \lambda_1 \hat{P}(t_i)$$

Trigram-HMMs (3)

Calculation of $\lambda_1, \lambda_2, \lambda_3$ for instance via deleted interpolation

$\lambda_1 = \lambda_2 = \lambda_3 = 0$
for each trigram t_1, t_2, t_3 with $f(t_1, t_2, t_3) > 0$:
    depending on the maximum of the values
    $x_1 = \frac{C(t_1,t_2,t_3)-1}{C(t_1,t_2)-1}$, $x_2 = \frac{C(t_2,t_3)-1}{C(t_2)-1}$ and $x_3 = \frac{C(t_3)-1}{N-1}$:
        case max $= x_1$: increment λ_3 by $C(t_1, t_2, t_3)$
        case max $= x_2$: increment λ_2 by $C(t_1, t_2, t_3)$
        case max $= x_3$: increment λ_1 by $C(t_1, t_2, t_3)$
normalize $\lambda_1, \lambda_2, \lambda_3$
(N is the overall number of tokens in the corpus)

Accuracy = (number of correctly tagged tokens) / (number of tokens)
with POS tagging is usually greater than 95 %.

A comparison of some state of the art POS taggers can be found here:
https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

Eisner, J. (2002). An interactive spreadsheet for teaching the forward-backward algorithm. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL*, pages 10–18.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall Series in Articial Intelligence. Pearson Education International, second edition edition.

Jurafsky, D. and Martin, J. H. (2017). Speech and language processing. an introduction to natural language processing, computational linguistics, and speech recognition. Draft of the 3rd edition.