

# Hidden-Markov-Modelle: Wortart-Tagging

Was berechnet ein Wortart-Tagger (Formel mit Argmax) ?  
Wofür stehen die jeweiligen Variablen?

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

**Aufgabe 1)** Leiten Sie die Formel für **Hidden-Markowmodelle** her, d.h. zeigen Sie, wie Sie von  $\arg \max_{t_1^n} p(t_1^n | w_1^n)$  zu  $\arg \max_{t_1^n} \prod_{i=1}^{n+1} p(t_i | t_{i-1}) p(w_i | t_i)$  kommen.

Geben Sie an, welche vereinfachenden Annahmen Sie dabei machen.

Erklären Sie, warum der Index  $i$  in der Produktformel bis  $n + 1$  läuft. (5 Punkte)

Gesucht ist die wahrscheinlichste Wortartfolge  $\hat{t}_1^n$  für eine gegebene Wortfolge  $w_1^n = w_1, \dots, w_n$

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

Die Konstante  $p(w_1^n)$  hat keinen Einfluss auf das Maximierungsergebnis.

Zerlegung in ein Produkt bedingter Wahrscheinlichkeiten:

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_1^{i-1}) \prod_{i=1}^{n+1} p(w_i | w_1^{i-1}, t_1^n)$$

Ähnlich wie bei den Markowmodellen wird ein Endetag  $t_{n+1} = \langle s \rangle$  und ein Endetoken  $w_{n+1} = \epsilon$  hinzugefügt, mit  $p(\epsilon | \langle s \rangle) = 1$ .

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_1^{i-1}) p(w_i | w_1^{i-1}, t_1^{n+1})$$

Wir machen nun die folgenden vereinfachenden Annahmen:

- Das Wortart-Tag  $t_i$  hängt nur von den  $k$  vorherigen Tags ab.
- Das Wort  $w_i$  hängt nur von seiner Wortart  $t_i$  ab.
- Die Wahrscheinlichkeiten sind unabhängig von der Position.

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

Für  $k = 2$  erhält man einen Trigramm-Tagger.

Dieses Modell heißt **Hidden-Markow-Modell**, weil die Zustände (Tags bzw. Tagpaare beim Trigramm-Tagger) nicht direkt beobachtbar sind.

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_{i-k}^{i-1}) p(w_i | t_i)$$

k= 1

p(NE VVFIN, Peter liest) =

p(NE|<s>) p(Peter|NE) \* p(VFIN|NE) p(liest|VFIN)  
 p(<s>|liest) p(""|<s>)

k= 2

p(NE VVFIN, Peter liest) = .....

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_{i-k}^{i-1}) p(w_i | t_i)$$

k= 2

p(NE VVFIN, Peter liest) = .....

i=	-1	0	1	2	3
t=	<s>	<s>	NN	VVFIN	<s>
w=	" "	" "	Peter	liest	" "

i= 1 , p(NN|<s>,<s>) p(Peter|NN)

i= 2 , p(VVFIN|<s>,NN) p(liest|VVFIN)

i= 3 , p(<s>|NN, VVFIN) p(" " | <s>)

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_{i-k}^{i-1}) p(w_i | t_i)$$

Bei einem HMM 1. Ordnung (k=1) gilt:

$$p(\text{NE VVFIN, Peter liest}) = p(\text{NE} | \langle s \rangle) p(\text{Peter} | \text{NE}) \cdot \\ p(\text{VVFIN} | \text{NE}) p(\text{liest} | \text{VVFIN}) \cdot \\ p(\langle s \rangle | \text{VVFIN}) p(\epsilon | \langle s \rangle)$$

Beim HMM 2. Ordnung (k=2) gilt:

$$p(\text{NE VVFIN, Peter liest}) = p(\text{NE} | \langle s \rangle, \langle s \rangle) p(\text{Peter} | \text{NE}) \cdot \\ p(\text{VVFIN} | \langle s \rangle, \text{NE}) p(\text{liest} | \text{VVFIN}) \cdot \\ p(\langle s \rangle | \text{NE, VVFIN}) p(\epsilon | \langle s \rangle)$$



**Aufgabe 3)** Geben Sie die allgemeine Formel an, mit der bei einem Hidden-Markow-Modell 2. Ordnung (d.h. einem Trigramm-HMM) die Wahrscheinlichkeit  $p(w_1^n, t_1^n)$  einer Wortfolge  $w_1^n$  mit der Tagfolge  $t_1^n$  berechnet wird. Was ist bzgl. Satzanfang und Satzende zu beachten?

Welche Wahrscheinlichkeiten müssen konkret für die Wortfolge “Es regnet” und die Tagfolge “PPER VVFIN” multipliziert werden? (3 Punkte)

**Aufgabe 3)** Geben Sie die allgemeine Formel an, mit der bei einem Hidden-Markow-Modell 2. Ordnung (d.h. einem Trigramm-HMM) die Wahrscheinlichkeit  $p(w_1^n, t_1^n)$  einer Wortfolge  $w_1^n$  mit der Tagfolge  $t_1^n$  berechnet wird. Was ist bzgl. Satzanfang und Satzende zu beachten?

Welche Wahrscheinlichkeiten müssen konkret für die Wortfolge “Es regnet” und die Tagfolge “PPER VVFIN” multipliziert werden? (3 Punkte)

$$p(w_1^n, t_1^n) = \prod_{i=1}^{n+1} p(t_i | t_{i-2}^{i-1}) p(w_i | t_i)$$

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_{i-k}^{i-1}) p(w_i | t_i)$$

k= 2

p(PPER VVFIN, Es regnet) = .....

i=	-1	0	1	2	3
t=	<s>	<s>	PPER	VVFIN	<s>
w=	<s>	<s>	Es	regnet	<s>

i= 1 , p(PPER|<s>) p(Es|PPER)

i= 2 , .....

i= 3 , .....

## Zwei Typen von Parametern

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

Die **Kontextwahrscheinlichkeiten** erfassen die syntaktischen Abhängigkeiten der Wortart-Tags vom linken Satzkontext.

Die **lexikalischen Wahrscheinlichkeiten** erfassen die syntaktischen Eigenschaften des aktuellen Wortes.

Spielt der rechte Kontext hier keine Rolle?

## Rechter Satzkontext: Beispiel 2

Ein HMM ist gegeben durch die Tabelle:

	A	B	$\langle s \rangle$	a	b	x	$\epsilon$
A	0.5	0	0.5	0.5	0	0.5	0
B	0	0.5	0.5	0	0.5	0.5	0
$\langle s \rangle$	0.5	0.5	0	0	0	0	1

mit  $p(\langle s \rangle | A) = 0.5$  und  $p(x | A) = 0.5$ .

- Die einzig mögliche Tagfolge der Tokenfolge xxxxxxa lautet AAAAAAA.
- Die einzig mögliche Tagfolge der Tokenfolge xxxxxx b lautet BBBBBBB.
- Bei allen anderen Tagfolgen ist die gemeinsame Wahrscheinlichkeit mit der Tokenfolge gleich 0.
- Hier besteht also eine Abhängigkeit von einem beliebig weit entfernten Token!

## Parameterschätzung

Zur Schätzung der Parameter braucht man ein **Trainingskorpus**, in dem jedes Wort mit seiner Wortart annotiert ist.

Man zählt die  $k+1$ -Gramm-Häufigkeiten  $f(t_1, \dots, t_{k+1})$  und die Tag-Wort-Häufigkeiten  $f(t, w)$  und schätzt dann die Wahrscheinlichkeiten wie folgt:

$$p(w|t) = \frac{f(t, w)}{\sum_{w'} f(t, w')} \quad \Leftrightarrow \quad p(t|t_1, \dots, t_k) = \frac{f(t_1, \dots, t_k, t)}{\sum_{t'} f(t_1, \dots, t_k, t')}$$

Beispiel:

$$p(\text{Haus}|\text{NN}) = \frac{f(\text{NN}, \text{Haus})}{\sum_{w'} f(\text{NN}, w')} \quad p(\text{NN}|\text{ART}) = \frac{f(\text{ART}, \text{NN})}{\sum_{t'} f(\text{ART}, t')}$$

Die Kontextwahrscheinlichkeiten müssen aber noch geglättet werden.

$$p(w|t) = \frac{f(t, w)}{\sum_{w'} f(t, w')} \quad \Rightarrow \quad p(t|t_1, \dots, t_k) = \frac{f(t_1, \dots, t_k, t)}{\sum_{t'} f(t_1, \dots, t_k, t')}$$

Wie werden  $p(\text{NN} | \langle s \rangle)$  und  $p(\text{Peter} | \text{NE})$  laut obiger Formel geschätzt?

$$\begin{aligned} p(\text{NN} | \langle s \rangle) &= f(\langle s \rangle, \text{NN}) / \text{Sum}_{t'} f(\langle s \rangle, t') \\ &= 20 / 22 \end{aligned}$$

$$\begin{aligned} p(\text{Peter} | \text{NE}) &= f(\text{NE}, \text{Peter}) / \text{Sum}_{w'} f(\text{NE}, w') \\ &= 3 / 3 + 2 \end{aligned}$$

$$f(\text{NE}, \text{Peter}) = 3$$

$$f(\text{NE}, \text{Lala}) = 2$$

$$f(\text{DT}, \text{the}) = 23$$

$$f(\langle s \rangle, \text{NN}) = 20$$

$$f(\langle s \rangle, \text{DT}) = 2$$

$$f(\text{DT}, \text{NN}) = 15$$

$$p(w|t) = \frac{f(t, w)}{\sum_{w'} f(t, w')} \quad \text{und} \quad p(t|t_1, \dots, t_k) = \frac{f(t_1, \dots, t_k, t)}{\sum_{t'} f(t_1, \dots, t_k, t')}$$

Wie werden  $p(\text{NN} | \langle s \rangle)$  und  $p(\text{Peter} | \text{NE})$  laut obiger Formel geschätzt?

$$p(\text{NE} | \langle s \rangle) = \frac{f(\langle s \rangle, \text{NE})}{\sum_{t'} f(\langle s \rangle, t')}$$

$$p(\text{Peter} | \text{NE}) = \frac{f(\text{NE}, \text{Peter})}{\sum_{w'} f(\text{NE}, w')}$$



$$p(w|t) = \frac{f(t, w)}{\sum_{w'} f(t, w')} \quad \circ \quad p(t|t_1, \dots, t_k) = \frac{f(t_1, \dots, t_k, t)}{\sum_{t'} f(t_1, \dots, t_k, t')}$$

Wie werden  $p(\text{NN} | \langle s \rangle \langle s \rangle)$  und  $p(\text{Peter} | \text{NE})$  laut obiger Formel geschätzt?

$$p(\text{NE} | \langle s \rangle \langle s \rangle) = \frac{f(\langle s \rangle \langle s \rangle, \text{NE})}{\sum_{w'} f(\langle s \rangle \langle s \rangle, w')}$$

$$p(\text{Peter} | \text{NE}) = \frac{f(\text{NE}, \text{Peter})}{\sum_{w'} f(\text{NE}, w')}$$

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

In der Praxis (Übung7), wird  $p(w|t)$  weiter modifiziert

$$p(w|t) = \frac{p(t|w)p(w)}{p(t)}$$

$$p(w_1^n, t_1^n) \propto \prod_{i=1}^{n+1} p(t_i | t_{i-2}, t_{i-1}) p(t_i | w_i) / p(t_i)$$

hier Formel für  $k=2$

Also in der Praxis werden die beiden  $p$  eigentlich nicht mit den Formeln unten geschätzt (sondern mit Backoff-Glättung)

$$p(w|t) = \frac{f(t, w)}{\sum_{w'} f(t, w')} \quad \text{↔} \quad p(t|t_1, \dots, t_k) = \frac{f(t_1, \dots, t_k, t)}{\sum_{t'} f(t_1, \dots, t_k, t')}$$

## Behandlung unbekannter Wörter

Kein manuell annotiertes Korpus ist so groß, dass es alle möglichen Wörter enthält.

Ein Wortart-Tagger muss daher immer mit **unbekannten Wörtern** umgehen können.

**Großschreibung** (Teil vs. teil) und **Wortendungen** (-keit, -lich) liefern wertvolle Information über die möglichen Wortarten eines Wortes.

Um diese Informationen nutzen zu können, müssen wir das Modell modifizieren. Wir wenden dazu das Bayes'sche Theorem auf die lexikalischen Wahrscheinlichkeiten an und erhalten:

$$p(w|t) = \frac{p(t|w)p(w)}{p(t)}$$

Die Konstante  $p(w)$  können wir bei der Suche nach der besten Tagfolge ignorieren (sofern die Tokenisierung eindeutig ist).

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

$$p(w_1^n, t_1^n) \propto \prod_{i=1}^{n+1} \frac{p(t_i | t_{i-2}, t_{i-1}) p(t_i | w_i)}{p(t_i)}$$

- Kontextwahrscheinlichkeiten  $p(t|t', t'')$

Diese Wahrscheinlichkeiten sollen mit Kneser-Ney-Glättung geglättet werden. Sie müssen hier nur die relativen Häufigkeiten  $p^*(t|t', t'')$ ,  $p^*(t|t'')$  und  $p^*(t)$  (vgl. Übung 3) berechnen.  $p(t|t', t'')$  wird erst im eigentlichen Taggerprogramm berechnet.

$$p(t|w) = r(t|w) + \alpha(w) p_{suff}(t|suffix(w))$$

mit  $r(t|w) = \max(0, \frac{f(w, t) - \delta}{\sum_{t'} f(w, t')})$

- Apriori-Tagwahrscheinlichkeiten  $p(t) = f(t) / \sum_{t'} f(t')$

## Behandlung unbekannter Wörter

Die Apriori-Wahrscheinlichkeit  $p(t) = f(t)/N$  ( $N$  = Korpusgröße) der Tags kann leicht aus dem Trainingskorpus geschätzt werden.

Die bedingte Wahrscheinlichkeit  $p(t|w)$  kann anhand des Wortsuffixes (z.B. der Länge  $l$ ) geschätzt werden:

$$p(t|w) = p(t|a_1 a_2 \dots a_n) \approx p_{\text{suff}}(t|a_{n-l} \dots a_n)$$

$$p(\text{NN}|\text{schwärzlich}) \approx p_{\text{suff}}(\text{NN}|\text{rzlich}) \quad \text{falls } l = 5$$

Die Suffix-Tag-Wahrscheinlichkeit  $(\alpha$  ist hier eine beliebige Buchstabenfolge.)

$$p_{\text{suff}}(t|s_1 \dots s_l) = \frac{f_{\text{suff}}(s_1 \dots s_l, t)}{\sum_{t'} f_{\text{suff}}(s_1 \dots s_l, t')} \quad f_{\text{suff}}(s_1 \dots s_l, t) = \sum_{w=\alpha s_1 \dots s_l} f(w, t)$$

wird aus Trainingsdaten geschätzt und mit einem Backoff-Verfahren geglättet.

Für groß- und kleingeschriebene Wörter schätzt man am besten separate Parameter. Dafür kann man einfach einen Buchstaben (z.B. "G" für Groß- und "K" für Kleinschreibung) an das Wortsuffix anhängen:  $p_{\text{suff}}(\text{NN}|\text{rzlichK})$

$$p(w|t) = \frac{p(t|w)p(w)}{p(t)}$$

$$p(t|w) = r(t|w) + \alpha(w) p_{\text{suff}}(t|\text{suffix}(w))$$

mit  $r(t|w) = \max(0, \frac{f(w,t) - \delta}{\sum_{t'} f(w,t')})$

↗

$$p(\text{NN} | \text{Möglichkeit}) = r(\text{NN} | \text{Möglichkeit}) + \text{backoff\_factor}(\text{Möglichkeit}) p_{\text{suff}}(\text{NN} | \text{suffix}(\text{Möglichkeit}))$$

$$r(\text{NN} | \text{Möglichkeit}) = f(\text{Möglichkeit}, \text{NN})\text{-discount} / \text{Sum} \dots$$

$$p_{\text{suff}}(\text{NN} | \text{suffix}(\text{Möglichkeit})) = p_{\text{suff}}(\text{NN} | \text{keit})$$

$$\text{suffix len} = 4$$

$$p_{\text{suff}}(\text{NN} | \text{keit}) = r(\text{NN} | \text{keit}) + \text{backoff}(\text{keit}) * p_{\text{suff}}(\text{NN} | \text{\_eit})$$

$$r(\text{NN} | \text{keit}) = f(\text{keit}, \text{NN})\text{-dis} / \text{Sum}_{t'} f(\text{keit}, t')$$

$$p(t|w) = r(t|w) + \alpha(w) p_{\text{suff}}(t|\text{suffix}(w))$$

mit  $r(t|w) = \max(0, \frac{f(w,t) - \delta}{\sum_{t'} f(w,t')})$

$$p(\text{NN} | \text{Möglichkeit}) = r(\text{NN} | \text{Möglichkeit}) + \text{backoff\_factor}(\text{Möglichkeit}) p_{\text{suff}}(\text{NN} | \text{suffix}(\text{Möglichkeit}))$$

$$r(\text{NN} | \text{Möglichkeit}) = \frac{f(\text{Möglichkeit}, \text{NN}) - \delta}{\sum_{t'} f(\text{Möglichkeit}, t')}$$

$$p(\text{NN} | \text{suffix}(\text{Möglichkeit})) = p(\text{NN} | \text{keit})$$

$$p(\text{NN} | \text{keit}) = r(\text{NN} | \text{keit}) + \alpha(\text{keit}) p(\text{NN} | \text{eit})$$

$$r(\text{NN} | \text{keit}) = \frac{f(\text{keit}, \text{NN}) - \delta}{\sum_{t'} f(\text{keit}, t')}$$

$$p(\text{NN} | \text{eit}) = \dots$$

$$p(t|w) = r(t|w) + \alpha(w) p_{suff}(t|suffix(w))$$

mit  $r(t|w) = \max(0, \frac{f(w,t) - \delta}{\sum_{t'} f(w,t')})$

?

Möglichkeit: NN

Möglichkeit: PP

Wahrscheinlichkeit: NN

Ausbildung: NN

essen: V

Essen: NN

Hunger: NN

Aufmerksamkeit: NN

gelbe: ADJ

wahrscheinlich: ADJ

vermutlich: ADJ

$$r(\text{NN} | \text{Möglichkeit}) = \frac{f(\text{Möglichkeit}, \text{NN}) - \text{discount}}{f(\text{Möglichkeit}, \text{NN}) + f(\text{Möglichkeit}, \text{PP})}$$

suffix len = 4

suffix(Möglichkeit) = keit

**ohne Backoff**

$$p_{suff}(\text{NN} | \text{keit}) = \frac{f(\text{keit}, \text{NN})}{f(\text{keit}, \text{NN}) + f(\text{keit}, \text{PP})}$$

$$f(\text{keit}, \text{NN}) = f(\text{Möglichkeit}, \text{NN}) + f(\text{Wahrscheinlichkeit}, \text{NN}) + f(\text{Aufmerksamkeit}, \text{NN})$$

$$p_{suff}(t|s_1 \dots s_l) = \frac{f_{suff}(s_1 \dots s_l, t)}{\sum_{t'} f_{suff}(s_1 \dots s_l, t')}$$

$$f_{suff}(s_1 \dots s_l, t) = \sum_{w = \alpha s_1 \dots s_l} f(w, t)$$



## Übung

$$p_{suff}(t|s_1 \dots s_l) = \frac{f_{suff}(s_1 \dots s_l, t)}{\sum_{t'} f_{suff}(s_1 \dots s_l, t')}$$

$$f_{suff}(s_1 \dots s_l, t) = \sum_{w=\alpha s_1 \dots s_l} f(w, t)$$

Möglichkeit: NN

Wahrscheinlichkeit: ADJ

Ausbildung: NN

essen: V

Essen: NN

endlich: PP

endlich: ADJ

Aufmerksamkeit: NN

gelbe: ADJ

wahrscheinlich: ADJ

vermutlich: ADJ

r(ADJ | endlich) =

suffix len = 4

suffix(endlich) =

**ohne Backoff**

p\_suff(ADJ | endlich) =

$$p(t|w) = \frac{f(w, t) - \delta_6}{\sum_{t'} f(w, t')} + \alpha(w)p(t|a_{n-4}^n g(w))$$

$$p(t|b_1^k g) = \frac{f(b_1^k g, t) - \delta_k}{\sum_{t'} f(b_1^k g, t')} + \alpha(b_1^k g)p(t|b_2^k g) \text{ für } 0 < k < 5$$

$$p(t|g) = \frac{f(g, t)}{\sum_{t'} f(g, t')}$$

**Wie berechnet man p\_suff( NN | keitg) Mit Backoff ?**

$$\begin{aligned} p\_suff( NN | \text{keitg}) &= f(\text{keitg}, NN) - \text{discount} / f(\text{keitg}, NN) + f(\text{keitg}, PP) \\ &\quad + \text{backoff}(\text{keitg}) \\ &\quad * p\_suff(NN | \text{eitg}) \end{aligned}$$

$$\begin{aligned} p\_suff(NN | \text{eitg}) &= f(\text{eitg}, NN) - \text{discount} / f(\text{eitg}, NN) + f(\text{eitg}, PP) + f(\text{eitg}, \dots) \\ &\quad + \text{backoff}(\text{eitg}) \\ &\quad * p\_suff(NN | \text{itg}) \end{aligned}$$

$$p\_suff(NN | \text{itg}) = \dots$$

...

$$p\_suff(NN | g) = f(g, NN) / f(g, NN) + f(g, PP) + f(g, V) + f(g, \dots)$$

## Beispiel zur Berechnung der lexikalischen Wahrscheinlichkeiten

für Suffixlänge  $l=5$ :

$$p(\text{NN}|\text{schwärzlich}) = r(\text{NN}|\text{schwärzlich}) + \alpha(\text{schwärzlich}) ( \\ r_{\text{suffix}}(\text{NN}|\text{zlichK}) + \alpha_{\text{suffix}}(\text{zlichK}) ( \\ r_{\text{suffix}}(\text{NN}|\text{lichK}) + \alpha_{\text{suffix}}(\text{lichK}) ( \\ r_{\text{suffix}}(\text{NN}|\text{ichK}) + \alpha_{\text{suffix}}(\text{ichK}) ( \\ r_{\text{suffix}}(\text{NN}|\text{chK}) + \alpha_{\text{suffix}}(\text{chK}) ( \\ r_{\text{suffix}}(\text{NN}|\text{hK}) + \alpha_{\text{suffix}}(\text{hK}) p(\text{NN}|\text{K}) ) ) ) ) ) )$$

Wörter mit weniger als  $l$  Buchstaben werden am Anfang mit Leerzeichen aufgefüllt werden, bevor das Suffix extrahiert wird.

Wende Bayes Theorem hier an

$$p(w|t) = \frac{p(t|w)p(w)}{p(t)}$$

$$p(w|t) = \frac{p(t|w)p(w)}{p(t)}$$

Wie berechnet man  $p(t)$  ?

$$p(w|t) = \frac{p(t|w)p(w)}{p(t)}$$

Wie berechnet man  $p(t)$  ?

$$p(w|t) = \frac{p(t|w)p(w)}{p(t)}$$

Die Apriori-Wahrscheinlichkeit  $p(t) = f(t)/N$  ( $N \hat{=}$  Korpusgröße) der Tags kann leicht aus dem Trainingskorpus geschätzt werden.

- Apriori-Tagwahrscheinlichkeiten  $p(t) = f(t) / \sum_{t'} f(t')$

Um  $p(t)$  berechnen zu können, brauchen wir die Häufigkeiten aller Tags.

Z.B.  $f(V) : 120$   
 $f(PP) : 50$   
 $f(DET) : 60$

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

Was wird gemacht, um mit unbekannten Tags bei der Berechnung von  $p(\text{tag}, \text{tag\_context})$  umzugehen?



$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

- Kontextwahrscheinlichkeiten  $p(t|t', t'')$

Diese Wahrscheinlichkeiten sollen mit Kneser-Ney-Glättung geglättet werden. Sie müssen hier nur die relativen Häufigkeiten  $p^*(t|t', t'')$ ,  $p^*(t|t'')$  und  $p^*(t)$  (vgl. Übung 3) berechnen.  $p(t|t', t'')$  wird erst im eigentlichen Taggerprogramm berechnet.

Was ist die Formel für  $p(t|t', t'')$  mit Backoff-Glättung?

Wie wird  $p^*(t|t', t'')$  geschätzt?

$$P(t|t', t'') = P^*(t|t', t'') + \alpha(t', t'') P(t|t'')$$

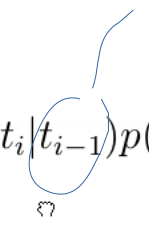
$$P^*(t|t', t'') = \frac{f(t', t'', t)}{\sum_{t'''} f(t', t'', t''')}$$

$$P(t|t'') = P^*(t|t'') + \alpha(t'') P(t)$$

$$P^*(t|t'') = \frac{f(t'', t)}{\sum_{t'''} f(t'', t''')}$$

$$P(t) = \frac{f(t)}{\sum_{t'} f(t')} = \frac{f(t)}{\text{Anzahl aller Tage}}$$

**Aufgabe 5)** Geben Sie an, mit welchen Umformungsschritten man die hier gezeigte Formel für das Hidden-Markov-Modell herleiten kann.

$$\arg \max_{t_1^n} p(t_1^n | w_1^n) = \dots = \arg \max_{t_1^n} \prod_{i=1}^{n+1} p(t_i | t_{i-1}) p(w_i | t_i)$$


(3 Punkte)

**Aufgabe 5)** Geben Sie an, mit welchen Umformungsschritten man die hier gezeigte Formel für das Hidden-Markov-Modell herleiten kann.

$$\arg \max_{t_1^n} p(t_1^n | w_1^n) = \dots = \arg \max_{t_1^n} \prod_{i=1}^{n+1} p(t_i | t_{i-1}) p(w_i | t_i)$$

(3 Punkte)

Gesucht ist die wahrscheinlichste Wortartfolge  $\hat{t}_1^n$  für eine gegebene Wortfolge  $w_1^n = w_1, \dots, w_n$

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

Die Konstante  $p(w_1^n)$  hat keinen Einfluss auf das Maximierungsergebnis.

Zerlegung in ein Produkt bedingter Wahrscheinlichkeiten:

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_{i-1}^{i-1}) \prod_{i=1}^{n+1} p(w_i | w_1^{i-1}, t_1^{i-1})$$

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_1^{i-1}) p(w_i | w_1^{i-1}, t_1^{i-1})$$

Wir machen nun die folgenden vereinfachenden Annahmen:

- Das Wortart-Tag  $t_i$  hängt nur von den  $k$  vorherigen Tags ab.
- Das Wort  $w_i$  hängt nur von seiner Wortart  $t_i$  ab.
- Die Wahrscheinlichkeiten sind unabhängig von der Position.

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

Für  $k = 2$  erhält man einen Trigramm-Tagger.

Dieses Modell heißt **Hidden-Markow-Modell**, weil die Zustände (Tags bzw. Tagpaare beim Trigramm-Tagger) nicht direkt beobachtbar sind.

**Aufgabe 3)** Das Hidden-Markow-Modell (HMM) ist durch die folgende Formel charakterisiert:

$$\hat{t}_1^n = \arg \max_{t_1^n} \prod_{i=1}^{n+1} p(t_i | t_{i-k}, \dots, t_{i-1}) p(w_i | t_i)$$

Erklären Sie

- wofür  $w_i$  und  $t_i$  genau stehen
- welche Bedeutung  $k$  hat
- wieso das Produkt bis  $i = n + 1$  berechnet wird
- welche Abhängigkeiten die beiden bedingten Wahrscheinlichkeiten jeweils beschreiben
- ob  $w_i$  von  $w_{i-1}$  statistisch unabhängig ist
- ob  $w_i$  von  $w_{i-1}$  statistisch unabhängig ist, wenn  $t_i$  bekannt ist

Geben Sie außerdem an, wie bei einem Bigramm-Tagger die Wahrscheinlichkeit der getaggten Wortfolge *es/PPER regnet/VVFIN* konkret definiert ist. (4 Punkte)

**Aufgabe 3)** Das Hidden-Markow-Modell (HMM) ist durch die folgende Formel charakterisiert:

$$\hat{t}_1^n = \arg \max_{t_1^n} \prod_{i=1}^{n+1} p(t_i | t_{i-k}, \dots, t_{i-1}) p(w_i | t_i)$$

Erklären Sie

- wofür  $w_i$  und  $t_i$  genau stehen  $w_i$  = wort an Position i  
 $t_i$  = Tag an Position i
- welche Bedeutung  $k$  hat  $k$  = Anzahl von Tagkontexte / Ordnung vom Markovmodell
- wieso das Produkt bis  $i = n + 1$  berechnet wird an der Position  $n+1$  wird  $\langle s \rangle$  hinzugefügt, damit alle WK zu 1 summiert

$\Rightarrow$  Es werden  $k$  Grenzsymbole  $\langle s \rangle$  am Anfang und eines am Ende hinzugefügt:

$$a_{-1} = a_0 = a_{n+1} = \langle s \rangle$$

Die Grenzsymbole am Anfang sind notwendig, damit  $p(a_1 | a_{-1}, a_0)$  definiert ist.

Ohne das Grenzsymbol am Ende würden die Wken aller Zeichenfolgen zu mehr als 1 summieren.

- welche Abhängigkeiten die beiden bedingten Wahrscheinlichkeiten jeweils beschreiben

Tag  $t$  ist abhängig von  $k$  vorherigen Tags.

Wort  $w$  an Position  $i$  ist abhängig von Tag  $t$  an dem gleichen Position

- ob  $w_i$  von  $w_{i-1}$  statistisch unabhängig ist
- ob  $w_i$  von  $w_{i-1}$  statistisch unabhängig ist, wenn  $t_i$  bekannt ist

Ich bin mir nicht sicher, ich glaube, Nein für beide Fragen

**Aufgabe 4)** Wie kann ein HMM-Tagger am besten mit unbekannten Wörtern umgehen? Erklären Sie, welche Änderungen an der obigen Formel des Modelles dazu notwendig sind. (3 Punkte)



**Aufgabe 4)** Wie kann ein HMM-Tagger am besten mit unbekannten Wörtern umgehen? Erklären Sie, welche Änderungen an der obigen Formel des Modelles dazu notwendig sind. (3 Punkte)

$$p(w|t) = \frac{p(t|w)p(w)}{p(t)}$$

$$\hat{t}_1^n = \arg \max_{t_1^n} \prod_{i=1}^{n+1} p(t_i | t_{i-k}^{i-1}) \frac{p(t_i | w_i)}{p(t_i)}$$

$$p(t|w) = r(t|w) + \alpha(w) p_{\text{suffix}}(t|\text{suffix}(w))$$

$$\text{mit } r(t|w) = \max(0, \frac{f(w,t) - \delta}{\sum_{t'} f(w,t')})$$

**Aufgabe 5)** Geben Sie die Formeln an, mit denen Sie die Kontext-Wahrscheinlichkeiten  $p(t|t', t'')$  und die lexikalischen Wahrscheinlichkeiten  $p(w|t)$  für einen HMM-Wortart-Tagger bei Verwendung einer **Maximum-Likelihood**-Schätzung berechnen. (2 Punkte)

**Aufgabe 5)** Geben Sie die Formeln an, mit denen Sie die Kontext-Wahrscheinlichkeiten  $p(t|t', t'')$  und die lexikalischen Wahrscheinlichkeiten  $p(w|t)$  für einen HMM-Wortart-Tagger bei Verwendung einer **Maximum-Likelihood**-Schätzung berechnen. (2 Punkte)

$$p(w|t) = \frac{f(t, w)}{\sum_{w'} f(t, w')} \quad p(t|t_1, \dots, t_k) = \frac{f(t_1, \dots, t_k, t)}{\sum_{t'} f(t_1, \dots, t_k, t')}$$

↗

Beispiel:

$$p(\text{Haus}|\text{NN}) = \frac{f(\text{NN}, \text{Haus})}{\sum_{w'} f(\text{NN}, w')} \quad p(\text{NN}|\text{ART}) = \frac{f(\text{ART}, \text{NN})}{\sum_{t'} f(\text{ART}, t')}$$

**Aufgabe 2)** Wie lauten die Formeln für die Berechnung der Wahrscheinlichkeiten  $p(t_3|t_1, t_2)$ , wenn Sie eine Maximum-Likelihood-Schätzung (= relative Häufigkeiten) verwenden? (2 Punkte)

**Aufgabe 2)** Wie lauten die Formeln für die Berechnung der Wahrscheinlichkeiten  $p(t_3|t_1, t_2)$ , wenn Sie eine Maximum-Likelihood-Schätzung (= relative Häufigkeiten) verwenden? (2 Punkte)

$$p(t_3|t_1, t_2) = \frac{f(t_1, t_2, t_3)}{\sum_{t'} f(t_1, t_2, t')}$$

**Aufgabe 3)** Wie lauten die Formeln zur Schätzung von **ungeglätteten Werten** für die Parameter eines HMM aus den Trainingsdaten (sog. Maximum-Likelihood-Estimate)? Geben Sie an, was mit den Variablen bezeichnet wird, die Sie verwenden. (2 Punkte)

---

**Aufgabe 3)** Wie lauten die Formeln zur Schätzung von **ungeglätteten Werten** für die Parameter eines HMM aus den Trainingsdaten (sog. Maximum-Likelihood-Estimate)? Geben Sie an, was mit den Variablen bezeichnet wird, die Sie verwenden. (2 Punkte)

$$p(w|t) = \frac{f(t, w)}{\sum_{w'} f(t, w')} \quad p(t|t_1, \dots, t_k) = \frac{f(t_1, \dots, t_k, t)}{\sum_{t'} f(t_1, \dots, t_k, t')}$$

Beispiel:

$$p(\text{Haus}|\text{NN}) = \frac{f(\text{NN}, \text{Haus})}{\sum_{w'} f(\text{NN}, w')} \quad p(\text{NN}|\text{ART}) = \frac{f(\text{ART}, \text{NN})}{\sum_{t'} f(\text{ART}, t')}$$

**Aufgabe 3)** Wie lautet die Formel zur Berechnung der Wahrscheinlichkeit  $p(t_3|t_1, t_2)$ , wenn Sie eine interpolierte Backoff-Schätzung mit Absolute Discounting verwenden?

\_\_\_\_\_ (3 Punkte)



**Aufgabe 3)** Wie lautet die Formel zur Berechnung der Wahrscheinlichkeit  $p(t_3|t_1, t_2)$ , wenn Sie eine interpolierte Backoff-Schätzung mit Absolute Discounting verwenden?  
(3 Punkte)

$$p(w_i|w_{i-k}^{i-1}) = \frac{\max(0, f(w_{i-k}^i) - \delta_k)}{f(w_{i-k}^{i-1})} + \alpha(w_{i-k}^{i-1}) p(w_i|w_{i-k+1}^{i-1})$$

Auch hier gilt:  $f(w_{i-k}^{i-1}) = \sum_w f(w_{i-k}^{i-1} w)$

$$p(t_3|t_1, t_2) = \frac{f(t_1, t_2, t_3) - \delta}{f(t_1, t_2)} + \alpha(t_1, t_2) p(t_3|t_2)$$

$$\downarrow$$

$$= \sum_t f(t_1, t_2, t)$$

**Aufgabe 5)** Wie glätten Sie die **Kontextwahrscheinlichkeiten** eines HMMs am besten? Geben Sie die Formel für das Glättungsverfahren an. (2 Punkte)

**Aufgabe 5)** Wie glätten Sie die **Kontextwahrscheinlichkeiten** eines HMMs am besten? Geben Sie die Formel für das Glättungsverfahren an. (2 Punkte)

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

Mit Backoff

$$p(t_i | t_{i-k}^{i-1}) = \frac{f(t_{i-k}^i) - \delta_k}{f(t_{i-k}^{i-1})} + \alpha(t_{i-k}^{i-1}) p(t_i | t_{i-k+1}^{i-1})$$

$$= \sum_t f(t_{i-k}^{i-1}, t)$$

**Aufgabe 1)** Geben Sie an, wie bei einem Bigramm-HMM-Tagger die Wahrscheinlichkeit einer Wortfolge  $w_1, \dots, w_n$  mit den Tags  $t_1, \dots, t_n$  definiert ist. Was ist an den Satzgrenzen zu beachten? (3 Punkte)

**Aufgabe 1)** Geben Sie an, wie bei einem Bigramm-HMM-Tagger die Wahrscheinlichkeit einer Wortfolge  $w_1, \dots, w_n$  mit den Tags  $t_1, \dots, t_n$  definiert ist. Was ist an den Satzgrenzen zu beachten? (3 Punkte)

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} \underbrace{p(t_i | t_{i-k}^{i-1})}_{\text{Kontextwahrsch.}} \underbrace{p(w_i | t_i)}_{\text{lexikalische Wk.}}$$

$$P(t_1, \dots, t_n, w_1, \dots, w_n) = \prod_{i=1}^{n+1} P(t_i | t_{i-2}, \dots, t_{i-1}) P(w_i | t_i)$$

**Aufgabe 2)** Welche Art von Daten brauchen Sie, um die Parameter des HMMs **über-**  
**wacht** zu trainieren? (1 Punkt)

↗

**Aufgabe 2)** Welche Art von Daten brauchen Sie, um die Parameter des HMMs **über-**  
**wacht** zu trainieren? (1 Punkt)

Zur Schätzung der Parameter braucht man ein **Trainingskorpus**, in dem jedes Wort mit seiner Wortart annotiert ist.

**Aufgabe 1)** Wie ist die Wahrscheinlichkeit der getaggten Wortfolge “The/DT man/NN slept/VBD” bei einem **HMM-Wortart-Tagger** 2. Ordnung (Trigramm-Tagger) definiert? (3 Punkte)



**Aufgabe 1)** Wie ist die Wahrscheinlichkeit der getaggten Wortfolge “The/DT man/NN slept/VBD” bei einem **HMM-Wortart-Tagger** 2. Ordnung (Trigramm-Tagger) definiert? (3 Punkte)

---

Beispiel

$$p(t_1^n, w_1^n) = \prod_{i=1}^{n+1} p(t_i | t_{i-k}^{i-1}) p(w_i | t_i)$$

Bei einem HMM 1. Ordnung (k=1) gilt:

$$\begin{aligned} p(\text{NE VVFIN, Peter liest}) &= p(\text{NE} | \langle s \rangle) p(\text{Peter} | \text{NE}) \cdot \\ &\quad p(\text{VVFIN} | \text{NE}) p(\text{liest} | \text{VVFIN}) \cdot \\ &\quad p(\langle s \rangle | \text{VVFIN}) p(\epsilon | \langle s \rangle) \end{aligned}$$

Beim HMM 2. Ordnung (k=2) gilt:

$$\begin{aligned} p(\text{NE VVFIN, Peter liest}) &= p(\text{NE} | \langle s \rangle, \langle s \rangle) p(\text{Peter} | \text{NE}) \cdot \\ &\quad p(\text{VVFIN} | \langle s \rangle, \text{NE}) p(\text{liest} | \text{VVFIN}) \cdot \\ &\quad p(\langle s \rangle | \text{NE}, \text{VVFIN}) p(\epsilon | \langle s \rangle) \end{aligned}$$

**Aufgabe 1)** Geben Sie die allgemeine Formel an, mit der bei einem Hidden-Markow-Modell 2. Ordnung (d.h. einem Trigramm-HMM) die Wahrscheinlichkeit  $p(w_1^n, t_1^n)$  einer Wortfolge  $w_1^n$  mit der Tagfolge  $t_1^n$  berechnet wird. Was ist bzgl. Satzanfang und Satzende zu beachten?

Welche Wahrscheinlichkeiten müssen konkret für die Wortfolge “Es regnet” und die Tagfolge “PPER VVFIN” multipliziert werden? (5 Punkte)