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 $w_1^n = w_1, ..., w_n$

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

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) = …..

k= 2
p(NE VVFIN, Peter liest) = …..

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

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

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

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

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

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)

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	ϵ
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 xxxxxxb 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 $f(t_1, ..., t_{k+1})$ und die Tag-Wort-Häufigkeiten $f(t, w)$ und schätzt dann die Wahrscheinlichkeiten wie folgt:

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

Beispiel:

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

Die Kontextwahrscheinlichkeiten müssen aber noch geglättet werden.

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

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

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

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

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

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

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

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

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

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

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

- 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

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

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:

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

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

Die Suffix-Tag-Wahrscheinlichkeit (α ist hier eine beliebige Buchstabenfolge.)

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

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: $p_{suff}(\text{NN}|\text{rzlichK})$

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

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

Suffix len = 4

p(NN| Möglichkeit) = r(NN| Möglichkeit) + backoff_factor(Möglichkeit) p_suff(NN| keit)

r(NN| Möglichkeit) =

Berechne p_suff(NN| keit) ohne Backoff
p_suff(PP| keit) = f_suff (keit, PP) / sum_t' von f_suff(keit, t')

f_suff(keit, PP) = sum_w von f(w mit keit am Ende, PP)

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

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

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

Suffix len = 4

p (V | disambiguieren) =

Berechne p_suff(tag| suffix) ohne Backoff
p_suff(…. | …..) =

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

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

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(NN | Möglichkeit) = f(Möglichkeit, NN) - discount
/ f(Möglichkeit, NN) + f(Möglichkeit, PP)

suffix len = 4
suffix(Möglichkeit) = keit

ohne Backoff
p_suff(NN | keit) = f(keit, NN) / f(keit,NN) + f(keit,PP)

f(keit,NN) = f(Möglichkeit, NN) +f(Wahrscheinlichkeit, NN)
+ f(Aufmerksamkeit, NN)

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

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

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) =

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

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

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

p_suff(NN | keit) Mit Backoff

p_suff(NN | keit) = f(keit, NN) -discount / f(keit,NN) + f(keit,PP)
+ backoff(keit)
* p_suff(NN| _eit)

p_suff(NN| _eit) = f(eit, NN) -discount / f(eit, NN) + f(eit, PP) + f(eit, …)
+ backoff(eit)
* p_suff(NN| _it)

….

Wende Bayes Theorem hier an

$$p(w|t) =$$

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

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

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

- 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.

Behandlung unbekannter Tags

Neben unbekannten Wörtern stellen auch ungesehene Tags bekannter Wörter ein Problem dar (bspw. wenn das englische Wort "indicate" nur als Infinitiv aber nicht als finites Verb aufgetaucht ist.)

Die suffixbasierten Tag-Wahrscheinlichkeiten geben meist auch den ungesehenen Tags eine kleine Wahrscheinlichkeit.

Daher ist es sinnvoll, wortbasierte und suffixbasierte Wahrscheinlichkeiten in einem Backoff-Modell zu kombinieren:

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

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

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

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

(3 Punkte)

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

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

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

Aufgabe 1) Geben Sie an, wie bei einem Bigramm-HMM-Tagger die Wahrscheinlichkeit einer Wortfolge $w1, ..., w_n$ mit den Tags $t_1, ..., t_n$ definiert ist. Was ist an den Satzgrenzen zu beachten? (3 Punkte)

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

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) 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)