Maschinelles Übersetzen natürlicher Sprachen

4. Übungsblatt

2016-11-17

Aufgabe 1 (Yamada-Knight-Modell)
Gegeben seien die folgenden Parameter des Yamada-Knight-Modells:

original order $\mathcal{R}(w)$	reordering ρ_w	$r(\rho_w \mid \mathcal{R}(w))$
PRP VB1 VB2	PRP VB1 VB2	0.074
	PRP VB2 VB1	0.723
	VB1 PRP VB2	0.061
	VB1 VB2 PRP	0.037
	VB2 PRP VB1	0.083
	VB2 VB1 PRP	0.021
VB TO	VB TO	0.251
	TO VB	0.749
TO NN	TO NN	0.107
	NN TO	0.893
⋮	⋮	⋮

y	$n_2(y)$
ha	0.219
ta	0.131
wo	0.099
no	0.094
ni	0.080
te	0.078
ga	0.062
⋮	⋮
desu	0.0007
⋮	⋮

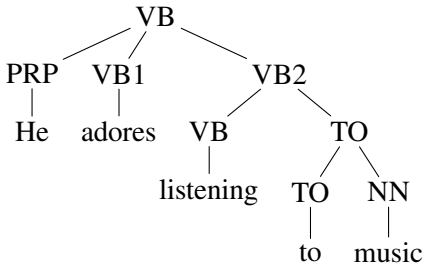
$n_1(x \mid \mathcal{I}(w))$	TOP VB	VB VB	VB PRP	VB TO	TO TO	TO NN	…
$x =$ no	0.735	0.687	0.344	0.709	0.900	0.800	…
$x =$ left	0.004	0.061	0.004	0.030	0.003	0.096	…
$x =$ right	0.260	0.252	0.652	0.261	0.007	0.104	…

$\mathcal{T}(w)$	adores		he		i	
$t(\tau_w \mid \mathcal{T}(w))$	daisuki	1.000	kare	0.952	NULL	0.471
			NULL	0.016	watasi	0.1111
			nani	0.005	kare	0.055
			da	0.003	shi	0.021
			shi	0.003	nani	0.020
			⋮	⋮	⋮	⋮

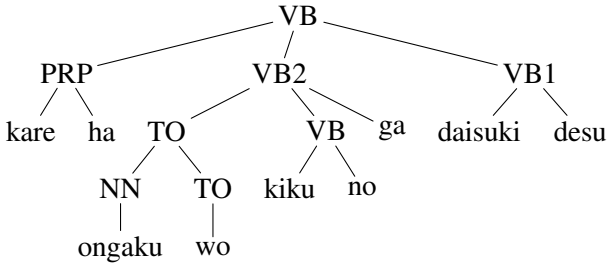
$\mathcal{T}(w)$	listening		music		to		...
$t(\tau_w \mid \mathcal{T}(w))$	kiku	0.333	ongaku	0.900	ni	0.216	...
	kii	0.333	naru	0.100	NULL	0.204	
	mi	0.333			to	0.133	
					no	0.046	
					wo	0.038	
					⋮	⋮	

$$\text{wobei } n(v_w \mid \mathcal{I}(w)) = \begin{cases} n_1(\text{no} \mid \mathcal{I}(w)) & \text{if } v_w = \text{no}, \\ n_1(x \mid \mathcal{I}(w)) \cdot n_2(y) & \text{if } v_w = (x, y). \end{cases}$$

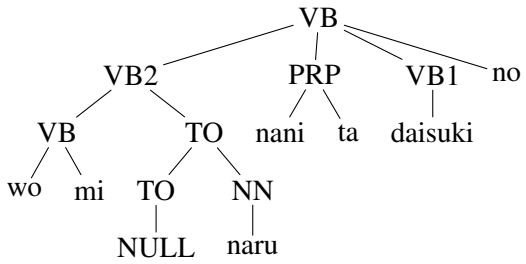
Wir betrachten die drei Bäume $t \in T_E$, sowie $s_1, s_2 \in T_F$:



t



s_1



s_2

Berechnen Sie die Übersetzungswahrscheinlichkeiten $P(s_1 \mid t)$ und $P(s_2 \mid t)$.

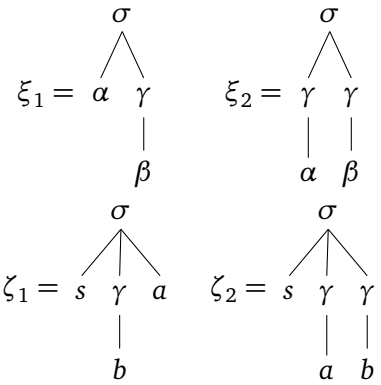
Ermitteln Sie nun die beste Übersetzung von t, also $\operatorname{argmax}_{s \in T_F} P(s \mid t)$. Kann die Umkehrung des Problems, also die Berechnung von $\operatorname{argmax}_{t \in T_E} P(s \mid t)$ aus einem gegebenen japanischen Baum s, auch so einfach gelöst werden? Warum (nicht)?

Aufgabe 2

Sei $\Sigma = \{\sigma, \gamma\}$ ein Alphabet. Wir nehmen ein bilinguales Baumkorpus $\mathcal{D} \subseteq \mathrm{U}_\Sigma(V_E) \times \mathrm{U}_\Sigma(V_F)$ zwischen zwei künstlichen Sprachen E und F an, deren Vokabular aus $V_E = \{\alpha, \beta\}$ bzw. aus $V_F = \{s, a, b\}$ bestehen soll.[1] Das Korpus ist recht klein; es setzt sich aus den zwei Baumpaaren

$$\mathcal{D} = \{(\xi_1, \zeta_1), (\xi_2, \zeta_2)\}$$

zusammen, wobei die Bäume von der folgenden Form sind:



[1] Dabei bezeichne $\mathrm{U}_\Sigma(A)$, gegeben ein Alphabet Σ und eine Menge A, die Menge der *ranglosen Bäume über Σ indiziert mit A*. Formal definieren wir $\mathrm{U}_\Sigma(A)$ als die kleinste Obermenge U von A, so dass für alle $\sigma \in \Sigma$, $k \in \mathbb{N}$ und $\xi_1, \ldots, \xi_k \in U$ auch $\sigma(\xi_1, \ldots, \xi_k) \in U$ gilt.

Nutzen Sie dieses Korpus zum Trainieren der Parameter eines Yamada-Knight-Übersetzungsmodells zwischen E und F nach dem in der Vorlesung vorgestellten Algorithmus! Wenden Sie dabei sinnvolle Vereinfachungen an, um die Anzahl der Parameter überschaubar zu halten.

Anmerkung: Es bietet sich an, die iterative Berechnungsvorschrift des Trainingsalgorithmus erst mathematisch auf das konkrete Beispiel zu instanziieren. Das wiederholte Anwenden dieser vereinfachten Berechnungsvorschrift lässt sich dann in einem kurzen Programm implementieren.