Schriftliche Prüfung zur Übung
Statistische Methoden in der maschinellen Sprachverarbeitung
WS 2013/14
Helmut Schmid

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)

Aufgabe 2)

Zeigen Sie, dass die Glättung mit dem Addiere-λ-Verfahren äquivalent zu einer Interpolation (gewichteten Mittelung) mit der uniformen Verteilung ist, d.h. zeigen Sie, dass folgende Gleichung gilt:

$$\frac{f(w_1^n)+\lambda}{N+B\lambda} = \mu\frac{f(w_1^n)}{N} + (1-\mu)\frac{1}{B} \quad \text{mit } \mu = \frac{N}{N+B\lambda}$$

(4 Punkte)

Aufgabe 3) Die Formel für die Backoff-Glättung mit Absolute Discounting lautet in der interpolierten Version:

$$p(a_k|a_1^{k-1}) = \frac{f(a_1^k)-\delta_k}{f(a_1^{k-1})} + \alpha(a_1^{k-1})p(a_k|a_2^{k-1})$$

Leiten Sie die Formel zur Berechnung von $\alpha(a_1^{k-1})$ her. (5 Punkte)

Aufgabe 4) Erklären Sie ausführlich, wie der Viterbi-Algorithmus beim Wortart-Taggen mit Hidden-Markow-Modellen die wahrscheinlichste Tagfolge berechnet. Geben Sie die Formeln zur Berechnung der Viterbiwahrscheinlichkeiten an. (4 Punkte)

Aufgabe 5) Schreiben Sie eine Programmfunktion "viterbi", welche die wahrscheinlichste Tagfolge für eine gegebene Wortfolge gemäß einem Bigramm-HMM berechnet. Der Funktion wird ein Array mit der Liste der Wörter als Argument übergeben.

Sie können annehmen, dass eine Funktion "lookup" bereits existiert, welche ein Wort als Argument nimmt und die Liste der Tags und ihre lexikalischen Wahrscheinlichkeiten $p(Wort|Tag)$ in einer von Ihnen gewünschten Datenstruktur zurückgibt. Sie können außerdem annehmen, dass eine Funktion "context_prob" existiert, welche zwei Wortart-Tags t_1 und t_2 als Argument nimmt und die Wahrscheinlichkeit $p(t_2|t_1)$ zurückliefert.

Überlegen Sie sich geeignete Datenstrukturen und schreiben Sie dann die Funktion. Sie können die Funktion in Perl, Python, C++ oder Java implementieren.

(12 Punkte)

(30 Punkte insgesamt)

Viel Erfolg!